A publishing infrastructure for AI-assisted academic authoring

This manuscript (permalink) was automatically generated from greenelab/manubot-gpt-manuscript@333107c on January 10, 2023.

Authors

- **Milton Pividori**
  0000-0002-3035-4403 · miltondp · miltondp
  Department of Genetics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA

- **Casey S. Greene** ✉
  0000-0001-8713-9213 · cgreene · GreeneScientist
  Center for Health AI, University of Colorado School of Medicine, Aurora, CO 80045, USA; Department of Biomedical Informatics, University of Colorado School of Medicine, Aurora, CO 80045, USA

✉ — Correspondence possible via GitHub Issues or email to Casey S. Greene <casey.s.greene@cuanschutz.edu>.

Abstract

Academics often communicate through scholarly manuscripts. These manuscripts describe new advances, summarize existing literature, or argue for changes in the status quo. Writing and revising manuscripts can be a time-consuming process. Large language models are bringing new capabilities to many areas of knowledge work. We integrated the use of large language models into the Manubot publishing ecosystem. Users of Manubot can run a workflow, which will trigger a series of queries to OpenAI's language models, produce revisions, and create a timestamped set of suggested revisions. Given the amount of time that researchers put into crafting prose, we expect this advance to radically transform the type of knowledge work that academics perform.

Introduction

The manuscript pre-dates the invention of printing by thousands of years, but the practice of producing exclusively scientific journals only started roughly 350 years ago [1]. The implementation of external peer review varies by journal but for many is less than 100 years old [2]. To date, most manuscripts have been written by humans or teams of humans working together to describe scholarly advances.

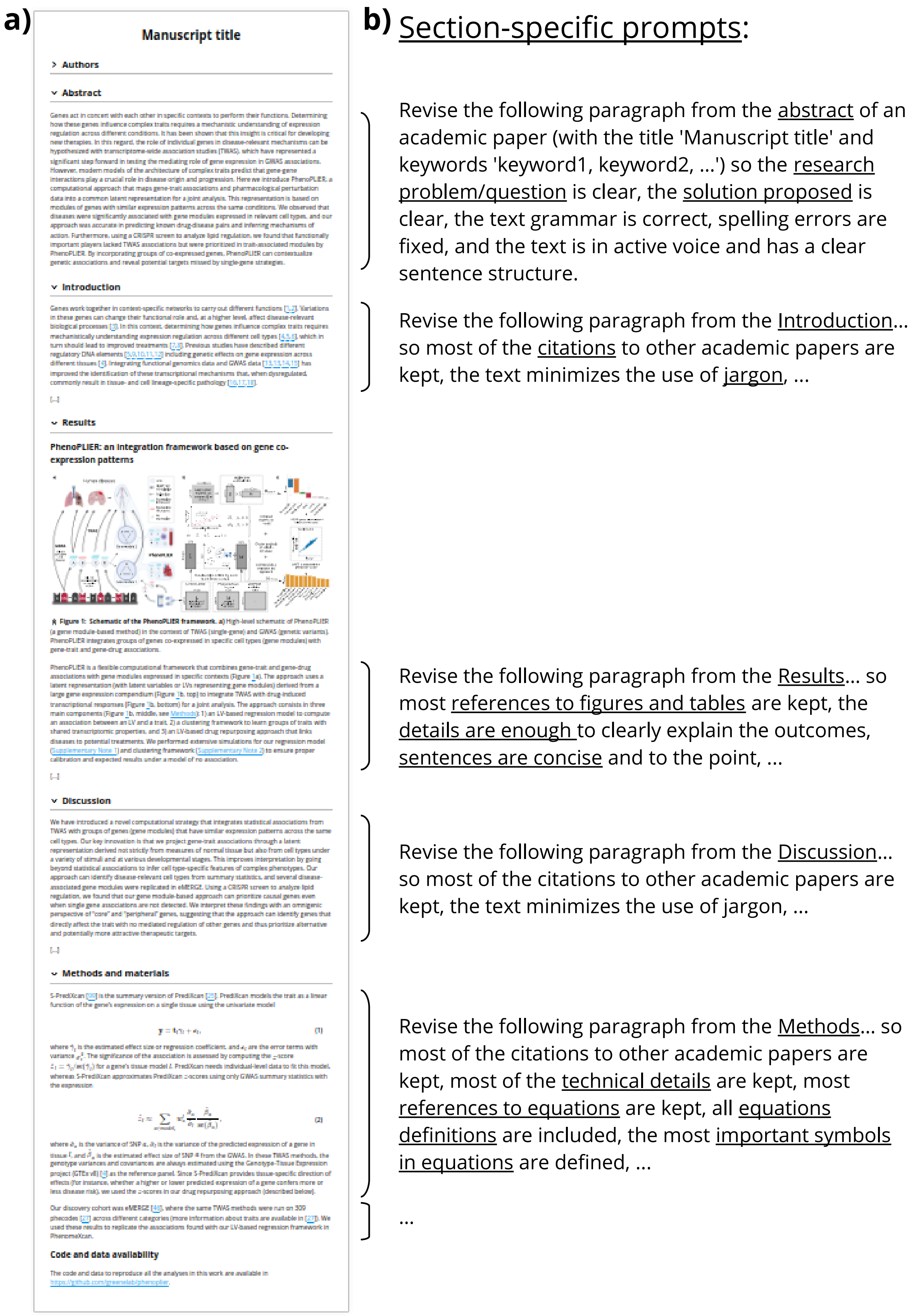
Modern scholarly manuscripts often describe new advances, summarize existing literature, or argue for changes in the status quo. However, writing and revising can be a time-consuming process. Academics can sometimes be long-winded in getting to key points, making writing more impenetrable to their audience [3].

Modern computing capabilities and the widespread availability of text, images, and other data on the internet has laid the foundation for artificial intelligence (AI) models with many parameters. Large language models, in particular, are opening the floodgates to new technologies with the capability to transform how society operates [4]. The GPT-3 model, with its 175 billion parameters, has demonstrated strong performance on many tasks [5].

We developed a software publishing platform that imagines a future where authors co-write their manuscripts with the support of large language models. We used, as a base, the Manubot platform for scholarly publishing [6]. Manubot was designed as an end-to-end publishing platform for scholarly writing for both individual and large-collaborative projects. It has been used for collaborations of approximately 50 authors writing hundreds of pages of text reviewing progress during the COVID19 pandemic [7]. We developed a new workflow that parses the manuscript, uses a large language model with section-specific custom prompts to revise the manuscript, and then creates a set of suggested changes to reach the revised state. Changes are presented to the user through the GitHub interface for author review and integration into the published document.

Implementing AI-based revision into the Manubot publishing ecosystem

Overview



Manuscript title

> Authors

Abstract

Genes act in concert with each other in specific contexts to perform their functions. Determining how these genes influence complex traits requires a mechanistic understanding of expression regulation across different conditions. It has been shown that this insight is critical for developing new therapies. In this regard, the role of individual genes in disease-relevant mechanisms can be hypothesized with transcriptome-wide association studies (TWAS), which have represented a significant step forward in testing the mediating role of gene expression in GWAS associations. However, modern models of the architecture of complex traits predict that gene-gene interactions play a crucial role in disease origin and progression. Here we introduce PhenoPLIER, a computational approach that maps gene-trait associations and pharmacological perturbation data into a common latent representation for a joint analysis. This representation is based on modules of genes with similar expression patterns across the same conditions. We observed that diseases were significantly associated with gene modules expressed in relevant cell types, and our approach was accurate in predicting known drug-disease pairs and inferring mechanisms of action. Furthermore, using a CRISPR screen to analyze lipid regulation, we found that functionally important players lacked TWAS associations but were prioritized in trait-associated modules by PhenoPLIER. By incorporating groups of co-expressed genes, PhenoPLIER can contextualize genetic associations and reveal potential targets missed by single-gene strategies.

Introduction

Genes work together in context-specific networks to carry out different functions [1,2]. Variations in these genes can change their functional role and, at a higher level, affect disease-relevant biological processes [3]. In this context, determining how genes influence complex traits requires mechanistically understanding expression regulation across different cell types [4,5,6], which in turn should lead to improved treatments [7,8]. Previous studies have described different regulatory DNA elements [5,9,10,11,12] including genetic effects on gene expression across different tissues [4]. Integrating functional genomics data and GWAS data [13,13,14,15] has improved the identification of these transcriptional mechanisms that, when dysregulated, commonly result in tissue- and cell lineage-specific pathology [16,17,18].

[...]

Results

PhenoPLIER: an integration framework based on gene co-expression patterns

Figure 1: Schematic of the PhenoPLIER framework. a) High-level schematic of PhenoPLIER (a gene module-based method) in the context of TWAS (single-gene) and GWAS (genetic variants). PhenoPLIER integrates groups of genes co-expressed in specific cell types (gene modules) with gene-trait and gene-drug associations.

PhenoPLIER is a flexible computational framework that combines gene-trait and gene-drug associations with gene modules expressed in specific contexts (Figure 1a). The approach uses a latent representation (with latent variables or LVs representing gene modules) derived from a large gene expression compendium (Figure 1b, top) to integrate TWAS with drug-induced transcriptional responses (Figure 1b, bottom) for a joint analysis. The approach consists in three main components (Figure 1b, middle, see Methods): 1) an LV-based regression model to compute an association between an LV and a trait, 2) a clustering framework to learn groups of traits with shared transcriptomic properties, and 3) an LV-based drug repurposing approach that links diseases to potential treatments. We performed extensive simulations for our regression model (Supplementary Note 1) and clustering framework (Supplementary Note 2) to ensure proper calibration and expected results under a model of no association.

[...]

Discussion

We have introduced a novel computational strategy that integrates statistical associations from TWAS with groups of genes (gene modules) that have similar expression patterns across the same cell types. Our key innovation is that we project gene-trait associations through a latent representation derived not strictly from measures of normal tissue but also from cell types under a variety of stimuli and at various developmental stages. This improves interpretation by going beyond statistical associations to infer cell type-specific features of complex phenotypes. Our approach can identify disease-relevant cell types from summary statistics, and several disease-associated gene modules were replicated in eMERGE. Using a CRISPR screen to analyze lipid regulation, we found that our gene module-based approach can prioritize causal genes even when single gene associations are not detected. We interpret these findings with an omnigenic perspective of "core" and "peripheral" genes, suggesting that the approach can identify genes that directly affect the trait with no mediated regulation of other genes and thus prioritize alternative and potentially more attractive therapeutic targets.

[...]

Methods and materials

S-PrediXcan [39] is the summary version of PrediXcan [25]. PrediXcan models the trait as a linear function of the gene's expression on a single tissue using the univariate model

$$\mathbf{y} = \mathbf{t}_l \gamma_l + \epsilon_l, \quad (1)$$

where $\hat{\gamma}_l$ is the estimated effect size or regression coefficient, and ϵ_l are the error terms with variance σ_ϵ^2. The significance of the association is assessed by computing the z-score $\hat{z}_l = \hat{\gamma}_l / \mathrm{se}(\hat{\gamma}_l)$ for a gene's tissue model l. PrediXcan needs individual-level data to fit this model, whereas S-PrediXcan approximates PrediXcan z-scores using only GWAS summary statistics with the expression

$$\hat{z}_l \approx \sum_{a \in model_l} w_a^l \frac{\hat{\sigma}_a}{\hat{\sigma}_l} \frac{\hat{\beta}_a}{\mathrm{se}(\hat{\beta}_a)}, \quad (2)$$

where $\hat{\sigma}_a$ is the variance of SNP a, $\hat{\sigma}_l$ is the variance of the predicted expression of a gene in tissue l, and $\hat{\beta}_a$ is the estimated effect size of SNP a from the GWAS. In these TWAS methods, the genotype variances and covariances are always estimated using the Genotype-Tissue Expression project (GTEx v8) [4] as the reference panel. Since S-PrediXcan provides tissue-specific direction of effects (for instance, whether a higher or lower predicted expression of a gene confers more or less disease risk), we used the z-scores in our drug repurposing approach (described below).

Our discovery cohort was eMERGE [40], where the same TWAS methods were run on 309 phecodes [27] across different categories (more information about traits are available in [27]). We used these results to replicate the associations found with our LV-based regression framework in PhenomeXcan.

Code and data availability

The code and data to reproduce all the analyses in this work are available in https://github.com/greenelab/phenoplier.

Section-specific prompts:

Revise the following paragraph from the abstract of an academic paper (with the title 'Manuscript title' and keywords 'keyword1, keyword2, ...') so the research problem/question is clear, the solution proposed is clear, the text grammar is correct, spelling errors are fixed, and the text is in active voice and has a clear sentence structure.

Revise the following paragraph from the Introduction... so most of the citations to other academic papers are kept, the text minimizes the use of jargon, ...

Revise the following paragraph from the Results... so most references to figures and tables are kept, the details are enough to clearly explain the outcomes, sentences are concise and to the point, ...

Revise the following paragraph from the Discussion... so most of the citations to other academic papers are kept, the text minimizes the use of jargon, ...

Revise the following paragraph from the Methods... so most of the citations to other academic papers are kept, most of the technical details are kept, most references to equations are kept, all equations definitions are included, the most important symbols in equations are defined, ...

...

Figure 1: AI-based revision applied on a Manubot-based manuscript. a) A manuscript (written with Manubot) with different sections. **b)** Section-specific prompts used to process each paragraph. If a paragraph belongs to a non-standard section, then a default prompt will be used to perform a basic revision only.

We implemented the AI-based revision infrastructure in Manubot [6]. Manubot is a tool for collaborative writing of scientific manuscripts. It utilizes version control and a continuous integration workflow to facilitate efficient and transparent collaboration among authors. Manubot integrates with popular version control platforms such as GitHub, allowing authors to easily track changes and collaborate on writing in real time. Additionally, Manubot automates the process of generating a formatted manuscript (such as HTML, PDF, DOCX; Figure 1a shows the HTML output), reducing the time and effort required for manuscript preparation and submission. Built on this modern and open paradigm, our AI-based revision software was built using GitHub Actions, which allows the user to easily trigger an automated revision task on the entire manuscript or specific sections of it.

When the user triggers the action, the manuscript is parsed by section and then by paragraph (Figure 1b), passed to the language model along with a set of custom prompts, returned, reformatted, and output. Our workflow then uses the GitHub API to generate a new pull request, allowing the user to review and modify the output before merging the changes into the manuscript. This workflow attributes text to either the human user or to the AI language model, which may be important if future legal decisions alter the copyright landscape around the outputs of generative models.

We used the OpenAI API for access to these models. Since this API incurs a cost with each run that depends on manuscript length, we implemented an workflow in GitHub Actions that can be manually triggered by the user. Our implementation allows users to tune the costs to their needs by allowing to select specific sections to be revised instead of the entire manuscript. Additionally, several model parameters can be adjusted to tune costs even further, such as the language model version (including Davinci and Curie, and potentially newly published ones), how much risk the model will take, or the “quality” of the completions. For instance, using Davinci models (the most complex and capable ones), the cost per run is under $0.50 for most manuscripts.

Implementation details

Our tools are comprised of Python scripts that perform the AI-based revision (https://github.com/greenelab/manubot-ai-editor) and a GitHub Actions workflow that integrates manuscript with Manubot. The user only needs to run the workflow by specifing the branch that will be revised and selecting the files/sections of the manuscript (optional), the language model to use (`text-davinci-003` by default) and the output branch name. As explained later, for more advanced users it is also possible change most of the tool’s behavior or the language model parameters.

When the workflow is triggered, it downloads the manuscript by cloning the specified branch. It revises all of the manuscript files, or only some of them if the user specifies a subset. Next, each paragraph in the file is read and submitted to the OpenAI API for revision. If the request is successful, the tool will write the revised paragraph in place of the original one using one sentence per line (which is the recommended format for the input text). If the request fails, the tool might try again (up to five times by default) if it is a common error (such as “server overloaded”) or a model specific error that requires to change some of its parameters. If the error cannot be handled or the maximum number of retries is reached, the original paragraph is written instead with an HTML comment at the top explaining the cause of the error. This allows the user to debug the problem and attempt to fix it if desired.

As shown in Figure 1b, each API request comprises a prompt (the instructions given to the model) and the paragraph to be revised. The prompt uses the manuscript title and keywords, so both have to be accurate for getting the best revision outcomes. The other key component to process a paragraph is its section. Some paragraphs are simpler to process than others. For instance, the abstract is a set of sentences with no citations, whereas a paragraph from the Introduction section has several references to other scientific papers. A paragraph in the Results section has fewer citations but many references to figures or tables, where enough details about the experiments must be provided to understand and interpret the outcomes. The Methods section is more dependent on the type of paper, but in general it has to provide technical details and sometimes mathematical formulas and equations. Therefore, we designed section-specific prompts, which we found led to the most useful suggestions. Figures and tables captions, as well as paragraphs that contain only one or two sentences and less than sixty words, are not processed and copied directly to the output file.

The section of a paragraph is automatically inferred from the file name using a simple strategy (such as if "introduction" or "methods" is part of the file name). If the tool fails to infer a section from the file, then the file will not be processed. If this happens, the user is still able to specify which section the file belongs to. The section could be a standard one (abstract, introduction, results, methods, or discussion) for which a specific prompt is used (Figure 1b), or a non-standard one for which a default prompt will be used to instruct the model to perform only a basic revision (`minimize the use of jargon, ensure text grammar is correct spelling errors are fixed, and the text has a clear sentence structure`).

Properties of language models

Our AI-based revision workflow uses text completion to process each paragraph, either using the completion endpoint or the new edits endpoint (which is currently in beta). We tested our tool using Davinci and Curie models, including `text-davinci-003`, `text-davinci-edit-001` and `text-curie-001`. Davinci models are the most powerful GPT-3 model, whereas Curie ones are less capable but faster and less expensive. Although the edits endpoints would be the ideal interface for our task, it is still in beta. Therefore, we mainly focused on the completion endpoint. All models can be fine-tuned using different parameters (see OpenAI - API Reference), and the most important ones can be easily adjusted using our tool.

Language models for text completion have a context length that indicates the limit of tokens they can process (tokens are common character sequences in text). This limit includes the size of the prompt and the paragraph, and the maximum number of tokens to generate for the completion (parameter `max_tokens`). For instance, the context length of Davinci models is 4,000, and 2,048 for Curie (see OpenAI - Models overview). For this reason, it is still not possible to use the entire manuscript as input, not even entire sections. Therefore, our AI-assisted revision software process each paragraph of the manuscript with section-specific prompts, as shown in Figure 1b. The advantage of this approach is the ability to process large manuscripts by processing small chunks of text. The main issue, however, is that the language model processes only a single paragraph from a section, potentially losing important context to produce a better output. Nonetheless, we find that the model still produces high-quality output (see Results). Additionally, since the goal of our tool is to revise a paragraph, by default we set the maximum number of tokens (parameter `max_tokens`) as twice the estimated number of tokens in the paragraph (one token approximately represents four characters, see OpenAI - Tokenizer). The tool automatically adjusts this parameter and performs the request again if a related error is returned by the API. The user can force the tool to either use a fixed value for `max_tokens` for all paragraphs, or change the fraction of maximum tokens based on the estimated paragraph size (two by default).

The language models used are stochastic: they will generate a different revision for the same input paragraph each time. This behavior can be changed by using the "sampling temperature" or "nucleus sampling" parameters (we use `temperature=0.5` by default). Although we selected default values that worked well across multiple manuscripts, these parameters can be changed by the user if necessary to make the model more deterministic. The user can also instruct the model to generate, for each paragraph, several completions and select the one with the highest log probability per token, what can improve the quality of the revision. Our proof-of-concept implementation generates only one completion (parameter `best_of=1`) to avoid potentially high costs for the user. Additionally, our workflow allows to process either the entire manuscript or individual sections. This allows to control costs more effectively while focusing on a single piece of text in which the user can run the tool several times and pick the prefered revised text.

Observations of AI-based revisions

Evaluation setup

We evaluated our AI-based revision workflow by testing different language models and manuscripts. For this, we used three different GPT-3 models from OpenAI: `text-davinci-003`, `text-davinci-edit-001`, and `text-curie-001`. The first two are based on the most capable Davinci models, (see [OpenAI - GPT-3 models](#)). The difference between them is that `text-davinci-003` is a production-ready model for the completion endpoint, whereas `text-davinci-edit-001` is used for the newly edits endopoint (in beta). The edits endpoint provides a more natural interface for the revision of manuscripts since it has two inputs: the instructions and the text to revise. This is different from the completion endpoint, where there is a single input that contains the instructions and the text to revise. Finally, we also selected `text-curie-001` because, in addition to being faster and chepear than Davinci models, it is defined as a “very capable” model by their authors (see [OpenAI - GPT-3 models](#)).

Table 1: Manuscripts used to evaluate the AI-based revision workflow. The title and keywords of a manuscript are used in prompts for revising paragraphs. IDs are used in the text to refer to them, and they link to their GitHub repositories.

Manuscript ID	Title	Keywords
CCC	An efficient not-only-linear correlation coefficient based on machine learning	correlation coefficient, nonlinear relationships, gene expression
PhenoPLIER	Projecting genetic associations through gene expression patterns highlights disease etiology and drug mechanisms	genetic studies, functional genomics, gene co-expression, therapeutic targets, drug repurposing, clustering of complex traits
Manubot-AI	A publishing infrastructure for AI-assisted academic authoring	manubot, artificial intelligence, scholarly publishing, software

Assessing the performance of an automated revision tool is not straightforward. For this reason, we used three manuscripts of our own authorship to be able to more accurately assess the quality of the revision (Table 1). The first two are existing manuscripts that were previously written, and the third one is this manuscript which was written and then revised using our tool before submission. The first manuscript describes the Clustermatch Correlation Coefficient (CCC) [8], a new correlation coefficient that was evaluated in transcriptomic data to find novel, potentially nonlinear relationships between gene pairs in the Genotype-Tissue Expression v8 (GTEx) project. The second manuscript describes PhenoPLIER [9], a framework that comprises three different methods to improve the interpretability of genetic studies of complex diseases. We refer to these two manuscripts as CCC and PhenoPLIER, respectively. CCC is in the field of computational biology, whereas PhenoPLIER is in the field of genomic medicine. CCC describes one computational method applied to one data type (correlation to gene expression). PhenoPLIER describes a framework that comprises three different approaches (regression, clustering and drug-disease prediction) using data from genome-wide and transcription-wide association studies (GWAS and TWAS), gene expression, and transcriptional responses to small molecule perturbations. Therefore, CCC provides has a simpler structure, whereas PhenoPLIER is a more complex manuscript with more figures and tables and a Methods section including equations for different methods. The third manuscript is this one, where we describe software that uses machine learning models for the automated revision of scientific manuscripts, and we refer to it as Manubot-AI. Manubot-AI provides an example with a simple structure and significantly less figures than the rest. It was written and revised using our tool before submission, which provides a more real AI-based revision use case. These three manuscripts allowed us to significantly improve and test our prompts, and we report these findings below.

We enabled the Manubot AI revision workflow in the GitHub repositories of the three manuscripts (CCC: `https://github.com/greenelab/ccc-manuscript`, PhenoPLIER: `https://github.com/greenelab/phenoplier_manuscript`, Manubot-AI: `https://github.com/greenelab/manubot-gpt-manuscript`). This added the `"AI-revision"` workflow to the "Actions" tab of each repository, which allows to be manually triggered by the user. Then, we ran the workflow on the three manuscripts using the three language models described above, producing one pull request (PR) per manuscript and model. These PRs (three per manuscript) can be accessed from the "Pull requests" tab from each repository, where they are titled *"GPT (MODEL) used to revise manuscript"* with *MODEL* being the identifier of the model used. PRs show the differences between the original text and the suggestions made by the AI-based revision tool. Below we discussed our findings based on these PRs using the language models across different sections of the manuscripts.

Performance of language models

We found that Davinci models, as expected, were superior than the Curie model for all manuscripts. The Curie model is described as "very capable", and it is faster and less expensive than Davinci models. However, as shown in the PRs generated using this model (titled `GPT (text-curie-001) used to revise manuscript`), the model was not able to produce acceptable revisions for any of the manuscripts. Most of its suggestions were not coherent with the original text in any of the sections.

Among Davinci models, we found that for `text-davinci-edit-001` (edits endpoint), the quality of the revisions was subjectively inferior to `text-davinci-003` (completion endpoint). In general, the model either did not produce a revision (such as for abstracts) or the suggested changes were minimal or did not improve the original text. In paragraphs from the introduction, for instance, this model failed to keep references to other scientific articles in CCC, and in PhenoPLIER it didn't produce a meaningful revision. This might be explained by the fact that the edits endpoint is still in beta.

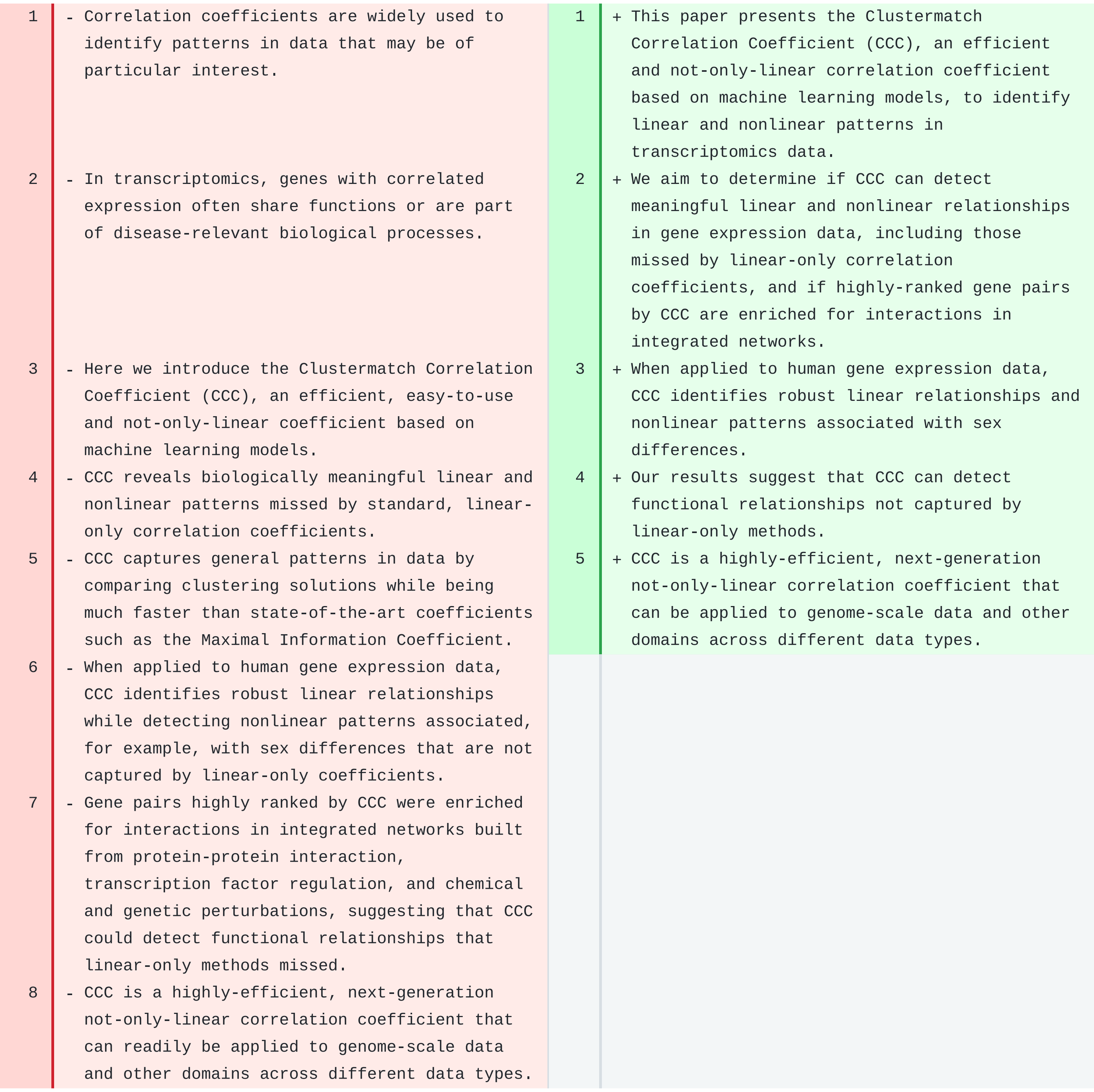
The `text-davinci-003` model produced the best results for all manuscripts and across the different sections. Since both `text-davinci-003` and `text-davinci-edit-001` are based on the same models, we only report the results of `text-davinci-003` below.

Revision of different sections

We inspected the PRs generated by the AI-based workflow, and highlight below some of the most interesting changes suggested by the tool across different section of the manuscripts. We show the differences between the original text and the revisions by the tool in a `diff` format (obtained from GitHub), where the original text is on the left, and the suggested one on the right. Line numbers were also included to more easily see the differences in length. When applicable, single words are also underlined and highlighted in colors to more clearly see the differences within a single sentence. In these cases, words underlined in red were removed by the tool, whereas words underlined in green were added and words not underlined were kept unchanged. The full diffs can be seen by inspecting the PRs for each manuscript and model, and then clicking on the "Files changed" tab.

Abstract

This is the revision for the abstract of CCC:



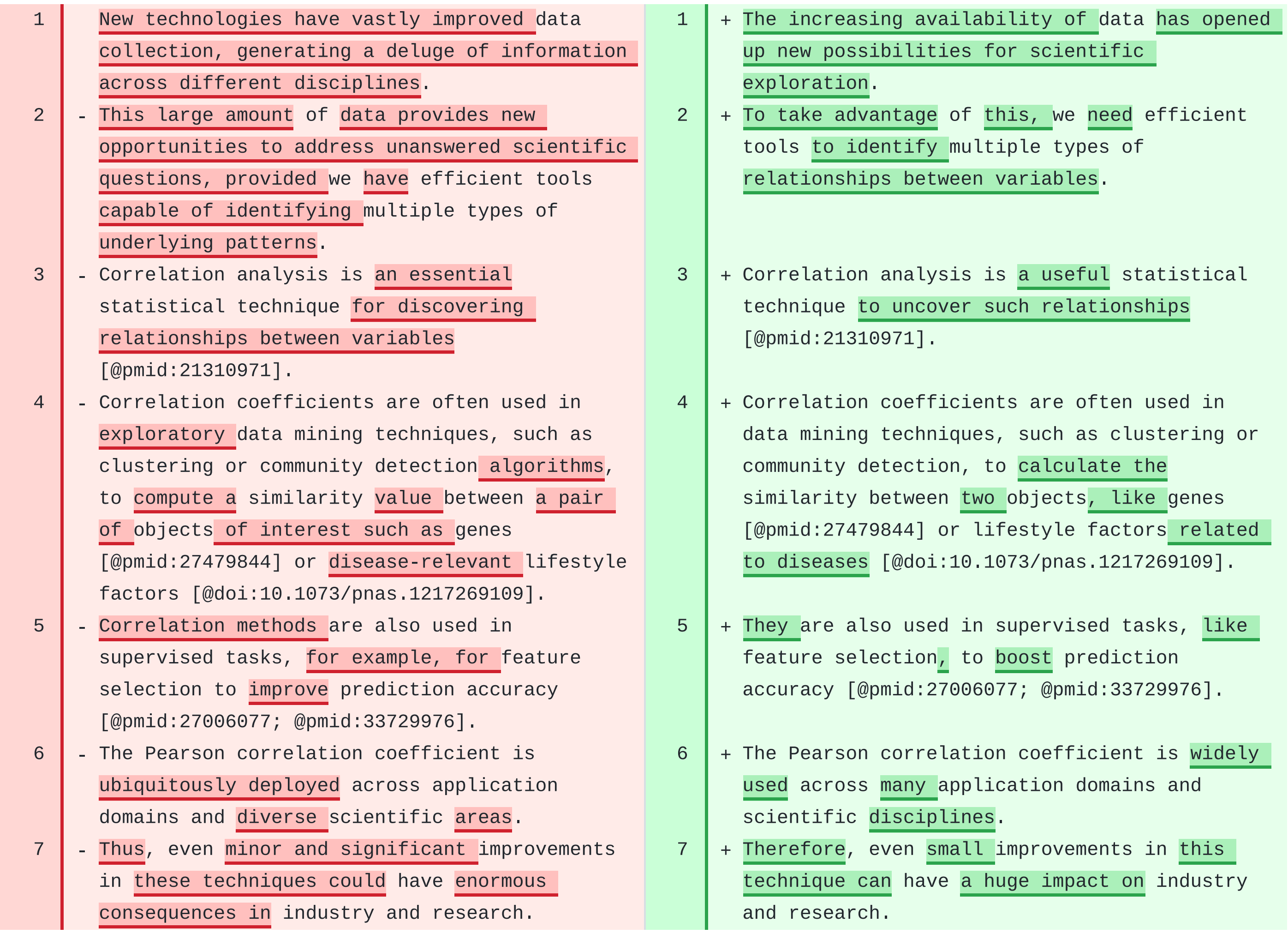
1	- Correlation coefficients are widely used to identify patterns in data that may be of particular interest.	1	+ This paper presents the Clustermatch Correlation Coefficient (CCC), an efficient and not-only-linear correlation coefficient based on machine learning models, to identify linear and nonlinear patterns in transcriptomics data.
2	- In transcriptomics, genes with correlated expression often share functions or are part of disease-relevant biological processes.	2	+ We aim to determine if CCC can detect meaningful linear and nonlinear relationships in gene expression data, including those missed by linear-only correlation coefficients, and if highly-ranked gene pairs by CCC are enriched for interactions in integrated networks.
3	- Here we introduce the Clustermatch Correlation Coefficient (CCC), an efficient, easy-to-use and not-only-linear coefficient based on machine learning models.	3	+ When applied to human gene expression data, CCC identifies robust linear relationships and nonlinear patterns associated with sex differences.
4	- CCC reveals biologically meaningful linear and nonlinear patterns missed by standard, linear-only correlation coefficients.	4	+ Our results suggest that CCC can detect functional relationships not captured by linear-only methods.
5	- CCC captures general patterns in data by comparing clustering solutions while being much faster than state-of-the-art coefficients such as the Maximal Information Coefficient.	5	+ CCC is a highly-efficient, next-generation not-only-linear correlation coefficient that can be applied to genome-scale data and other domains across different data types.
6	- When applied to human gene expression data, CCC identifies robust linear relationships while detecting nonlinear patterns associated, for example, with sex differences that are not captured by linear-only coefficients.		
7	- Gene pairs highly ranked by CCC were enriched for interactions in integrated networks built from protein-protein interaction, transcription factor regulation, and chemical and genetic perturbations, suggesting that CCC could detect functional relationships that linear-only methods missed.		
8	- CCC is a highly-efficient, next-generation not-only-linear correlation coefficient that can readily be applied to genome-scale data and other domains across different data types.		

The tool completely rewrote the text, where only the last sentence was mostly unchanged. The text was significantly shortened, although sentences are longer than the original ones which could make the abstract slightly harder to read. The revision removed the first two sentences that introduces correlation analyses and transcriptomics, and directly stated from the beginning the purpose of the manuscript. It also removed details about the method (line 5), and focused on the aims and the results obtained, ending with almost the same last sentence which suggest a more broad application of the coefficient to other data domains (as originally intended by the authors of CCC). However, none of the ideas suggested to be removed were critical, and all the main concepts are still present in the revised text.

The revised text for the abstract of PhenoPLIER was significantly shortened (from 10 sentences in the original, to only 3 in the revised version). However, in this case, important concepts (such as GWAS, TWAS, CRISPR) and a proper amount of background information were missing, producing a less informative abstract.

Introduction

This is the revision of the first paragraph of the introduction of CCC:



1	New technologies have vastly improved data collection, generating a deluge of information across different disciplines.	1	+ The increasing availability of data has opened up new possibilities for scientific exploration.
2	- This large amount of data provides new opportunities to address unanswered scientific questions, provided we have efficient tools capable of identifying multiple types of underlying patterns.	2	+ To take advantage of this, we need efficient tools to identify multiple types of relationships between variables.
3	- Correlation analysis is an essential statistical technique for discovering relationships between variables [@pmid:21310971].	3	+ Correlation analysis is a useful statistical technique to uncover such relationships [@pmid:21310971].
4	- Correlation coefficients are often used in exploratory data mining techniques, such as clustering or community detection algorithms, to compute a similarity value between a pair of objects of interest such as genes [@pmid:27479844] or disease-relevant lifestyle factors [@doi:10.1073/pnas.1217269109].	4	+ Correlation coefficients are often used in data mining techniques, such as clustering or community detection, to calculate the similarity between two objects, like genes [@pmid:27479844] or lifestyle factors related to diseases [@doi:10.1073/pnas.1217269109].
5	- Correlation methods are also used in supervised tasks, for example, for feature selection to improve prediction accuracy [@pmid:27006077; @pmid:33729976].	5	+ They are also used in supervised tasks, like feature selection, to boost prediction accuracy [@pmid:27006077; @pmid:33729976].
6	- The Pearson correlation coefficient is ubiquitously deployed across application domains and diverse scientific areas.	6	+ The Pearson correlation coefficient is widely used across many application domains and scientific disciplines.
7	- Thus, even minor and significant improvements in these techniques could have enormous consequences in industry and research.	7	+ Therefore, even small improvements in this technique can have a huge impact on industry and research.

The tool, again, significantly revised the text, producing a much better and more concise introductory paragraph. For example, the revised first sentence, in contrast with the original one, incorportes the ideas of “large datasets”, and the “opportunities/possibilities” for “scientific exploration” that they provide. Then the model generated a more concise and clear second sentence introducing the problem (“we need efficient tools” to find “multiple relationships” in these large datasets). The third sentence also nicely connects with the previous one. In comparison, the rest of the changes are minor but they still significantly improved the text. All references to scientific literature were kept using the correct Manubot format for citations, although our prompts never specify the format of the text (“Manubot” or “Markdown” is never mentioned). The rest of the sentences in this section of CCC were also correctly revised, and could be directly incorporated into the manuscript with minor or no further changes at all.

We also observed a high quality revision of the introdution of PhenoPLIER. For some paragraphs, however, the model failed to keep the format of citations, or the the models did not converge to a revised text.

Results

Discussion

Methods

Conclusions

We implemented AI-based models into publishing infrastructure. While most manuscripts have been written by humans, the process is time consuming and academic writing can be difficult to parse. We sought to develop a technology that academics could use to make their writing more understandable without changing the fundamental meaning. This work lays the foundation for a future where academic manuscripts are constructed by a process that incorporates both human and machine authors.

References

1. **A history of scientific & technical periodicals: the origins and development of the scientific and technical press, 1665-1790**
David A Kronick
Scarecrow Press (1976)
ISBN: 9780810808447

2. **The history of the peer-review process**
Ray Spier
Trends in Biotechnology (2002-08) https://doi.org/d26d8b
DOI: 10.1016/s0167-7799(02)01985-6 · PMID: 12127284

3. **How to write a first-class paper**
Virginia Gewin
Nature (2018-02-28) https://doi.org/ggh63n
DOI: 10.1038/d41586-018-02404-4

4. **Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models**
Alex Tamkin, Miles Brundage, Jack Clark, Deep Ganguli
arXiv (2021-02-05) https://arxiv.org/abs/2102.02503

5. **Language Models are Few-Shot Learners**
Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, … Dario Amodei
arXiv (2020-07-24) https://arxiv.org/abs/2005.14165

6. **Open collaborative writing with Manubot**
Daniel S Himmelstein, Vincent Rubinetti, David R Slochower, Dongbo Hu, Venkat S Malladi, Casey S Greene, Anthony Gitter
PLOS Computational Biology (2019-06-24) https://doi.org/c7np
DOI: 10.1371/journal.pcbi.1007128 · PMID: 31233491 · PMCID: PMC6611653

7. **An Open-Publishing Response to the COVID-19 Infodemic.**
Halie M Rando, Simina M Boca, Lucy D'Agostino McGowan, Daniel S Himmelstein, Michael P Robson, Vincent Rubinetti, Ryan Velazquez, Casey S Greene, Anthony Gitter
ArXiv (2021-09-17) https://www.ncbi.nlm.nih.gov/pubmed/34545336
PMID: 34545336 · PMCID: PMC8452106

8. **An efficient not-only-linear correlation coefficient based on machine learning**
Milton Pividori, Marylyn D Ritchie, Diego H Milone, Casey S Greene
Cold Spring Harbor Laboratory (2022-06-17) https://doi.org/gqcvbw
DOI: 10.1101/2022.06.15.496326

9. **Projecting genetic associations through gene expression patterns highlights disease etiology and drug mechanisms**
Milton Pividori, Sumei Lu, Binglan Li, Chun Su, Matthew E Johnson, Wei-Qi Wei, Qiping Feng, Bahram Namjou, Krzysztof Kiryluk, Iftikhar Kullo, … Casey S Greene
Cold Spring Harbor Laboratory (2021-07-06) https://doi.org/gk9g25
DOI: 10.1101/2021.07.05.450786