Richtlinie zum „Software-Sprint“

gget
– Genomische Datenbanken immer zur Hand

Schlussbericht

Zuwendungsempfänger:
Laura Lübbert

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS23S07 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der Autorin.

Kurze Darstellung der Aufgabenstellung und Motivation

Methoden der molekularen Genetik werden zunehmend preiswerter und effizienter. Ihre Nutzung in der akademischen und industriellen Wissenschaft wächst und hat in Gebieten wie der personalisierten Medizin auch im Alltag Einzug gehalten. Ein Beispiel ist single-cell RNA Sequencing, wobei die RNA-Sequenzen tausender Zellen an einem bestimmten Zeitpunkt gelesen werden. Dies erlaubt einen Einblick in den „Gemütszustand“ jeder einzelnen Zelle und was diese in Zukunft vorhat. Bei allem Potenzial bringt die Optimierung neuer Methoden jedoch Herausforderungen mit sich. Damit viele Millionen von RNA-Sequenzen interpretiert werden können, muss man diese auf die bekannten Gene zurückführen und herausfinden, welche Rolle sie in der Zelle spielen. Hierfür werden die Ergebnisse der RNA-Analyse mit bekannten DNA- Sequenzen, Genen, Proteinen und allen damit verbundenen Informationen verglichen. Solche Informationen sind zwar öffentlich verfügbar, aber über viele große, oft schwer zu navigierende online-Datenbanken verteilt. Manuell auf diese Informationen zuzugreifen ist mühsam und fehleranfällig.

gget ist ein Python- und Terminal-Programm, das den Abruf der Daten aus genomischen Datenbanken automatisiert, optimiert und beschleunigt. gget kombiniert MySQL, Application Programming Interface (API) und Web Data Extraction Methoden, mit denen schnell und zuverlässig relevante Informationen aus verschiedenen Datenbanken, zum Beispiel Ensembl, NCBI und UniProt, abgerufen werden können. gget integriert außerdem andere Open-Source Programme, etwa das Programm Muscle zum Vergleich von RNA- und DNA-Sequenzen und AlphaFold, für das Vorhersagen von Proteinstrukturen. Bestehende Programme werden als binäre Dateien automatisch zusammen mit gget installiert. Diese einzigartige Kombination erlaubt es gget Aufgaben auszuführen, die von bestehenden Programmen nicht unterstützt werden. Die aktuelle Quellcode Programmiersprache ist Python.

Ich habe im Mai 2022 die erste Version von gget mit neun Modulen auf Github und in einem bioRxiv paper (https://doi.org/10.1101/2022.05.17.492392) veröffentlicht. gget wurde von der Bioinformatik-Gemeinde mit Enthusiasmus begrüßt und seit seiner Veröffentlichung mehr als 50.000-mal heruntergeladen. Daher habe ich beschlossen gget weiter auszubauen, Datenbanken hinzuzufügen, und automatische Tests und Updates einzubauen, um zu jedem Zeitpunkt Präzision zu garantieren.

Meilensteine:

- gget's HTTP-Anfragen müssen an eventuelle Änderungen in der Struktur der Datenbanken angepasst werden. Hierzu soll
    - mit den folgenden Datenbanken Kontakt aufgenommen werden, um benötigte Anpassungen zu reduzieren: Ensembl, UniProt, NCBI, Enrichr, UCSB Genome Browser, Archs4, PDB.
    - automatisches Testen ausgebaut werden.
- (Github) Anfragen von Benutzern sollen eingebaut werden. Hierzu sollen
    - neue Module hinzugefügt / Daten von fehlenden und neuen Datenbanken, wie zum Beispiel UCSC Genome Browser, Chembl bioactivity database, gnomAD, genenames.org und GeneCards integriert werden.
    - bestehende Module verbessert werden. Zum Beispiel eine neue Funktion im Modul gget enrichr, die es Nutzern erlaubt, gegen eine Liste von Baseline Genen zu normalisieren.
- Als Open-Source-Projekt soll gget nicht nur für jeden erhältlich sein, sondern auch zu einem Gemeinschaftsprojekt wachsen, zu dem jeder durch Github pull requests beitragen kann.
    - Um die Qualität von gget weiterhin zu garantieren, müssen die Infrastruktur und Richtlinien für Community-Input ausgebaut werden. Dies soll die einfache Benutzungsweise, umfangreiche Tests, ausführliche Anleitung, und, wie in den Auflagen der Jury genannt, den Kontakt mit neuen Datenbanken auch in der Zukunft garantieren, während das Programm wächst.
    - Um die Übersetzung von gget in andere Programmiersprachen, zum Beispiel Rust, zu ermöglichen ohne große Abweichungen vom Originalprogramm, sollen die automatischen Tests in ein separates Github repository verlegt werden. Alle gget „Übersetzungen" können dann auf dieses zentrale Test repository zugreifen, um einheitliche Ergebnisse zu garantieren.
    - Für gget soll als Community-Projekt geworben werden, zum Beispiel durch gget Lehrveranstaltungen und Konferenzbesuche, mit dem Ziel, Community-Input zu ermutigen.

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint"

Die Existenz eines einfach zu bedienenden, Open-Source Programmes für den Zugriff auf Informationen, die auf viele große online-Datenbanken verteilt sind, ersetzt die manuelle online-Suche und zieht die Daten direkt in eine Programmierungsumgebung. Dies spart Zeit und reduziert die Fehlerrate, die mit manuellem Kopieren und Einfügen einhergeht. Außerdem erlaubt dies die Reproduzierbarkeit von Informationsflüssen und den daraus hervorgehenden Schlussfolgerungen, was der Reproduzierbarkeitskrise von wissenschaftlichen Publikationen entgegenwirkt. Zuletzt ist oft nicht bekannt, welche der Datenbanken relevante Informationen enthalten. Durch die Größenordnung von genomischen

Datenanalysen ist es praktisch unmöglich, immer alle Datenbanken manuell zu durchsuchen, wodurch manuelle Suchen oft voreingenommen sind und dazu tendieren, bereits bestehende Theorien zu bevorzugen. gget kombiniert automatisch Informationen aus vielen verschiedenen Datenbanken und macht das Vergleichen der Informationen aus verschiedenen Datenbanken zum Standard.

gget trägt zu den Zielen des Software-Sprints bei da es die Informationen auf genomischen Datenbanken zugänglicher macht und damit die Qualität von Wissenschaft mit großem gesellschaftlichem Einfluss wesentlich erhöht.

Ausführliche Darstellung der Ergebnisse

Alle oben gennanten Meilensteine wurden im Laufe der Projektzeit erreicht. Highlights:

- gget Webseite erneuert und auf Spanish übersetzt für erhöhte Zugänglichkeit: https://pachterlab.github.io/gget/
- Infrastruktur und Richtlinien für Community-Input ausgebaut: https://pachterlab.github.io/gget/en/contributing.html
- Lehrveranstaltungen organisiert, e.g. https://twitter.com/NeuroLuebbert/status/1684990102005768192?s=20
- Module erneuert / verbessert / ausgebaut:
    - gget enrichr
    - gget search
    - gget pdb
    - gget alphafold
    - gget info
- Neue Kollaborationen mit / Module für die folgenden Datenbanken:
    - gget cellxgene
    - gget gpt
    - gget elm

gget ist inzwischen zu einem großen Projekt geworden und, da ich im Moment die einzige Software-Betreuerin bin, war es wichtig die Aufgaben zu priorisieren. Hierbei war auch das Coaching hilfreich.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weiter-gehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?
Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Die Zielgruppe besteht aus (Bio-)Informatikern, Biologen, Medizinern, und Datenwissenschaftlern in allen Bereichen der genomischen Datenanalyse. Da *gget* inzwischen zu einem weltweiten Standard in der Analyse von genomischen Daten geworden ist, wird es weiterhin einen Strom von GitHub issues und pull requests geben. Da ich im Rahmen des Prototype fundings die community guidelines für gget ausgebaut habe, können Nutzer ihre Anfragen in Zukunft einfacher selber zu gget hinzufügen was die Arbeit der maintainer*innen reduziert. Es muss allerdings weiterhin die Qualität von gget garantiert werden und ich arbeite daran ein gget Team aufzubauen. Die intensieve Arbeit an gget hat mich sowohl bei meiner

persönlichen, als auch fachlichen Weiterentwicklung unterstützt: Durch die verschiedenen Bedürfnisse des gget Projekts habe ich viele Soft Skills und technische Fähigkeiten gelernt.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Während der Zusammenarbeit mit der Datenbank ELM kam relativ schnell heraus, dass deren Server nicht die Massen an gget Nutzen unterstützen können. Ich habe das Problem gelöst durch, in Zusammenarbeit mit dem ELM Team, das gget elm Modul komplett umzuschreiben, sodass es stattdessen die Ergebnisse lokal berechnet.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Interessenten können sich über die folgenden Links detailliert über gget informieren:
Webseite: https://pachterlab.github.io/gget/
GitHub: https://github.com/pachterlab/gget
Veröffentlichung: https://doi.org/10.1093/bioinformatics/btac836

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Ich hatte am Anfang des Projektes geplant insgesamt 480 Stunden an gget zu arbeiten. Diesen Plan habe ich tatsächlich relativ genau eingehalten, auch wenn einzelne Aufgaben etwas länger gedauert haben als ursprünglich gedacht (zum Beispiel das Übersetzen der gget Website ins Spanische).

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Einen großen Einfluss hatten die Nutzer Anfragen, die vor allem über GitHub (https://github.com/pachterlab/gget/issues) kommuniziert wurden. Es gab hier viele gute Ideen die ich im Laufe der Projektzeit übernommen habe und durch die gget besser geworden ist.