

DESIGN CONSIDERATIONS FOR GENOMICS TOOLS BASED ON CIRCULAR VISUALIZATION OF BIOLOGICAL DATA

CONSIDERAÇÕES DE DESIGN PARA FERRAMENTAS DE GENÔMICA BASEADAS EM VISUALIZAÇÃO CIRCULAR DE DADOS BIOLÓGICOS

João Lucas Dantas Silva, ID, (Universidade Federal do Pará, Pará, Brasil) – joao.dantas.silva@icen.ufpa.br

Colaboradores

Vinicius Augusto Carvalho de Abreu, ID, (Universidade Federal do Pará, Pará, Brasil) – VAbreu@ufpa.br

Marcelle Pereira Mota, 0000-0001-9226-9020, (Universidade Federal do Pará, Pará, Brasil) – mpmota@ufpa.br

ABSTRACT: With the increasing technological advances and reductions in the cost of genome sequencing, the field of bioinformatics has become one of the largest consumers of solutions for large amounts of data, which impacts the way solutions are developed in this field. This work seeks to map and raise hypotheses regarding the considerations and goals of developing visualization tools for biological data, focusing on tools that make use of circular graphs, in order to assist in the development of future tools and support the support of more existing tools, pointing out current problems and possible problems to come in the field. In this work, publication works for tools were analyzed, raising data in relation to the languages used for their development besides the application of structured analysis in heuristics of the Human-Computer Interaction field on their interfaces, bringing structured considerations under these analyses and proposing points and perspectives for future researches linked to software development methodology in the bioinformatics field.

Keywords: Human-Computer Interaction, Bioinformatics, Comparative Genomics, Data Visualization, Circular Visualization

RESUMO: A partir dos crescentes avanços tecnológicos e reduções no custo do sequenciamento de genomas, o campo da bioinformática torna-se um dos maiores consumidores de soluções para grandes quantidades de dados, o que acarreta em impactos no modo como desenvolvem-se soluções nesse campo. Este trabalho busca mapear e levantar hipóteses com relação às considerações e objetivos do desenvolvimento de ferramentas de visualização de dados biológicos, com foco em ferramentas que fazem uso de gráficos circulares, a fim de auxiliar no desenvolvimento de futuras ferramentas e apoiar o suporte de ferramentas mais existentes, apontando problemas atuais e possíveis problemas vindouros do campo. Neste trabalho, foram coletados e analisados trabalhos de publicação para ferramentas, levantando dados com relação às linguagens utilizadas para seus desenvolvimentos além da aplicação de análises estruturadas em heurísticas do campo de Interação Humano-Computador sobre suas interfaces, trazendo considerações estruturadas sob essas análises e propondo pontos e perspectivas para futuras pesquisas vinculadas a metodologia de desenvolvimento de *software* no campo da bioinformática.

Palavras-chave: Interação Humano-Computador, Bioinformática, Genômica Comparativa, Visualização de Dados, Visualização Circular

1. INTRODUÇÃO

A partir da publicação do sequenciamento e montagem do primeiro genoma completo, publicado em 1995 sobre a bactéria *Haemophilus influenzae* (FLEISCHMANN et al., 1995), a natureza dos dados associados a genômica costuma participar de relações com a área da computação, em grande parte pelo uso de técnicas computacionais para a realização de suas análises comparativas, tornando o campo da genômica um crescente representante do mercado associado a visualização de dados biológicos (“Biological Data Visualization Market Size & Share Report, 2030”, 2022).

Com isso em mente, considerando a redução de custos de sequenciamento para diversos organismos associados a recentes avanços tecnológicos, tem-se um cenário com fácil acesso a uma grande quantidade de dados provenientes da alta diversidade de genomas sequenciados, o que vem influenciando também na produção de um grande número de ferramentas desenvolvidas para visualização e realização de análises sobre esses dados.

Observando tais ferramentas, pode-se dizer que seus desenvolvimentos são realizados em grande parte no meio acadêmico, trazendo pouca consideração com relação a questões de acessibilidade e estudos de visualização de dados, devido a dificuldades associadas a tempo, verbas e à complexidade intrínseca de um trabalho interdisciplinar como tal. Porém, ao realizar uma busca na literatura, é possível identificar algumas convenções e técnicas comuns utilizadas por essas ferramentas, o que possibilita um olhar mais direcionado para o tratamento de dados em determinadas partes do processo de pesquisas comparativas.

Diante disso, este trabalho busca como principal contribuição a realização de análises sobre ferramentas recentes para a representação de dados biológicos por meio de visualização e plotagem circulares, com foco em comparação de genomas. A exemplo da Figura 1, trazendo comparações e considerações em relação a pontos de melhorias e ou novos aspectos de estudos para tais ferramentas. Tais considerações apresentadas serão pautadas em heurísticas e padrões estudados no campo de Interação Humano-Computador (IHC) para fim de análise da interface e de interação dessas ferramentas. Vale ressaltar também que, este trabalho foca em processos de análises de genômica comparativa realizadas do ponto de vista acadêmico, delimitando-se a ferramentas utilizadas no fluxo de trabalho de laboratórios acadêmicos, focados no campo da bioinformática e ferramentas com publicações recentes de cunho acadêmico.

Refletindo sobre o impacto esperado deste trabalho, pode-se dizer que, a associação de estudos direcionados a visualização de dados e outras questões do campo de IHC deve promover uma base teórica para uma melhor compreensão de aspectos como acessibilidade e usabilidade para o desenvolvimento de ferramentas ligadas a dados biológicos, possibilitando uma redução e facilitação de trabalhos em equipes, tanto em pesquisas acadêmicas mais abrangentes como em trabalhos mais específicos, a exemplo de trabalhos relacionados ao controle e tratamento de doenças. Com isso em mente, é possível afirmar que, ao realizar comparações e análises sustentadas por conceitos de IHC em ferramentas de visualização circular de genomas, teria-se não apenas considerações para futuros trabalhos de desenvolvimento de ferramentas dessa natureza como também um mapeamento de pontos de melhorias para tais ferramentas e demais estudos que utilizam dados biológicos como seu principal foco, sendo o resultado desse trabalho direcionado, porém não limitado, a tais técnicas de visualização de dados.

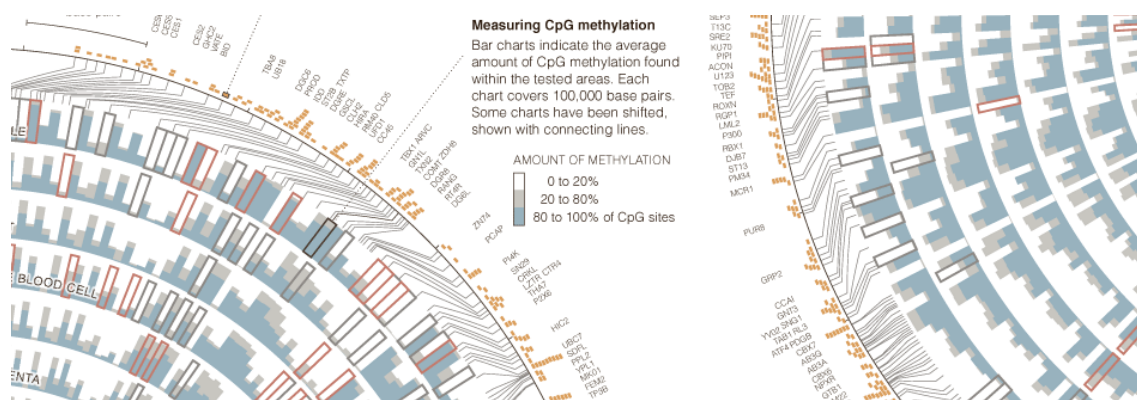


Figura 1 – Exemplo do uso de visualização circular de dados biológicos.

Fonte: Página de guias do site de distribuição da ferramenta, (“Guide to Application of Circos in Comparative Genomics and Visualizing Sequence Similarity and Conservation // CIRCOS Circular Genome Data Visualization”,).

2. METODOLOGIA

A dinâmica de seleção utilizada consistiu na coleta de ferramentas previamente identificadas e aplicadas para a realização de trabalhos de genômica comparativa. Além da coleta dos artigos de publicação destas ferramentas por meio do uso do motor de busca PubMed (“PubMed”, 2022), esse que permite o acesso ao banco de dados MEDLINE (“We make healthcare run better”, 2022), proporcionando fácil acesso pela sua natureza gratuita e integração com dados biológicos ou mesmo outros artigos citados nessas publicações, por fazer parte do ecossistema de ferramentas do National Institutes of Health (“National Institutes of Health (NIH)”, 2022), onde é possível acessar e coletar arquivos de genomas de organismo, por exemplo.

Para a busca das ferramentas, no motor de busca PubMed, foram utilizadas as seguinte palavras-chaves: “*circular visualization tool*” e “*circular genomics software*”.

Levando em conta a grande quantidade de resultados para pesquisas baseadas em palavras chaves e grande diversidade de ferramentas utilizadas em um único trabalho de genômica comparativa, foram definidos alguns critérios de seleção, dentre eles temos:

- Ferramentas com artigos publicados nos últimos 10 anos;
- Ferramentas com artigos gratuitos;
- Ferramentas com versões disponíveis de forma gratuitas;
- Ferramentas com versões funcionais para plataformas Windows ou Linux.

Além disso, vale ressaltar que trabalhos de *overview* do tema foram utilizados como material apoio para a fundamentação teórica utilizada neste trabalho, porém não como fonte de coleta para ferramentas, já que trabalhos dessa natureza acabam por trazer perspectivas estruturadas em parâmetros divergentes dos estabelecidos previamente, como no caso do trabalho “*Overview of Genomic Tools for Circular Visualization in the Next-generation Genomic Sequencing Era*”(PARVEEN; KHURANA; KUMAR, 2019), onde, por exemplo, há inclusão de ferramentas publicadas no período anterior a 10 anos atrás. No entanto, tal trabalho foi utilizado na formulação da discussão como referência teórica.

Do ponto de vista da avaliação das ferramentas, foi primeiramente definido os aspectos que seriam avaliados e utilizados para o mapeamento de problemas qualitativos que viriam a ser identificados em cada uma das ferramentas em análise, levando a escolha de um material utilizado como base inicial para a definição de tais aspectos. Dessa forma,

foi utilizado como material base o trabalho intitulado “Improving a human-computer dialogue” (MOLICH; NIELSEN, 1990), já que o mesmo é compreendido como base para o entendimento de erros compreendidos como básicos, facilitando a aplicação desses parâmetros em um campo tão diverso e recente como a bioinformática.

Realizando uma breve descrição das heurísticas utilizadas nesse trabalho, temos as seguintes:

- Visibilidade de estado no sistema: a interface deve apresentar de maneira clara o estado no qual o sistema se encontra para seu usuário;
- Correspondência entre o sistema e o mundo real: o sistema deve utilizar elementos e conceitos familiares aos seus usuários, seguindo convenções do mundo real, de forma que a informação apareça em uma ordem natural e lógica, conforme esperado pelos usuários;
- Controle e liberdade: a interface deve garantir a todo momento mecanismos de controle sobre ações realizadas por seus usuários, permitindo que o mesmo saia de um estado não desejado;
- Consistência e padronização: os elementos da interface devem seguir convenções e padronizações da plataforma ou do ambiente computacional, evitando ambiguidade para todos os seus elementos;
- Reconhecimento em vez de memorização: os elementos, os objetos, as ações e opções de uma interface devem ser visíveis e facilmente compreendidos por seus usuários, evitando a necessidade de memorização;
- Flexibilidade e eficiência de uso: a interface deve oferecer sistemas que facilitem a execução de determinadas tarefas por meio da rápida compreensão de funções almejadas por seus usuários. Além disso, a interface deve oferecer mecanismos de customizações para facilitar o uso de ações frequentes;
- Projeto estético e minimalista: a interface não deve conter informação que seja irrelevante ou raramente necessária.
- Prevenção de erros: buscar reduzir a ocorrência de erros quando possível;
- Reconhecer, diagnosticar e se recuperar: as mensagens de erro devem ser expressas em linguagem simples (sem códigos indecifráveis), indicar precisamente o problema e sugerir uma solução de forma construtiva;
- Ajuda e documentação: embora seja melhor que um sistema possa ser utilizado sem documentação, é necessário oferecer ajuda e documentação de alta qualidade. Tais informações devem ser facilmente encontradas, focadas na tarefa do usuário, enumerar passos concretos a serem realizados e não ser muito extensas.

Desse modo, as ferramentas coletadas, a partir de uma breve descrição e apresentação de características mais generalistas, passaram por análises e testes de uso realizados sobre suas interfaces gráficas, para aquelas que apresentavam a mesma. Nos casos de interação por linha de comando e pacotes para linguagem de programação, as análises resumiram-se a discussões mais generalistas, devido ao foco deste trabalho em interfaces gráficas.

Vale ressaltar também que, comumente a metodologia de avaliações heurísticas é aplicada com 3 avaliadores no mínimo, no entanto, devido a limitações de escopo, as avaliações heurísticas apresentadas neste trabalho foram realizadas por um avaliador apenas.

3. FERRAMENTAS

A partir da coleta das ferramentas e seus respectivos trabalhos de publicações, é possível compreender que o uso do termo ferramenta de visualização de dados, mesmo dentro do contexto apresentado e do recorte de coleta definido na metodologia, abrange diversos trabalhos e ferramentas com características e objetivos diferentes entre si, justificando e apontando para a necessidade da identificação de características que permitam um olhar que envolva, da melhor maneira, a visão dos seus desenvolvedores e que permita também uma melhor compreensão do público alvo em análises de cada uma dessas ferramentas.

A fim de facilitar tal processo, foi realizado um levantamento de características para cada uma das ferramentas coletadas, resultando no quadro apresentado ao longo desta seção, onde foram apresentadas características com relação à distribuição, suporte, disponibilização de código entre outras, tais informações estão dispostas no quadro 1.

Quadro 1 – Lista de ferramentas selecionadas.

Nome	Plataforma	LP	Ano	BI	DOI
CIG-P	Linux, Windows MacOS	Java	2014	GUI	10.1186/1471-2 105-15-344
CPGAVAS2	Web	Perl, Python	2019	GUI	10.1093/nar/gkz 345
ggtreeExtra	Linux, Windows MacOS	R	2021	Pacote	10.1093/molbev /msab166
Rcirc	Linux, MacOS	Perl, R	2020	Pacote	10.3389/fgene.2 020.00548
caOmicsV	Linux, Windows MacOS	R	2016	Pacote	10.1186/s12859 -016-0989-6
ClicO FS	Web	Ruby, JavaScript	2015	GUI	10.1093/bioinfo rmatcs/btv433
FuncTree	Web	JavaScript	2015	GUI	10.1371/journal .pone.0126967
karyoploteR	Linux, Windows MacOS	R	2017	Pacote	10.1093/bioinfo rmatcs/btx346
NG-Circos	Linux, Windows MacOS	JavaScript, Python	2020	CLI	10.1093/nargab/ lqaa069

OmicCircos	Linux, Windows MacOS	R	2014	Pacote	10.4137/CIN.S1 3495.
OGDraw	Web	Perl, D, JavaScript	2013	GUI	10.1093/nar/gkt 289
PACVr	Linux, Windows MacOS	R	2020	Pacote	10.1186/s12859 -020-3475-0
RCircos	Linux, Windows MacOS	R	2013	Pacote	10.1186/1471-2 105-14-244

Fonte: Elaboração própria (2022). LP – Linguagem de Programação; Ano – Ano de Publicação; BI – Base de Interface

Com esse levantamento inicial de informações, pode-se definir alguns grupos de ferramentas, baseados nessas informações, em conjunto com as descrições apresentadas por seus pesquisadores e desenvolvedores, permitindo que para as suas análises as mesmas fossem separadas com relação aos seus métodos de interface. Dessa forma, os métodos e consequentemente grupos de ferramentas foram divididos em Linha de Comando (CLI), Pacote e Interface Gráfica do Usuário (GUI), tais grupos serão melhor descritos em suas respectivas avaliações, sendo utilizados como base da divisão das análises realizadas.

4. AVALIAÇÕES E ANÁLISES

Com a elaboração da tabela apresentada anteriormente, os grupos selecionados serão apresentados a partir de uma análise e descrição de suas características para que, posteriormente, seja realizada e aplicada uma análise nas heurísticas de IHC citadas anteriormente. Dessa forma, para cada uma das ferramentas serão apresentadas as análises gerais e aspectos dos testes de uso, em conjunto com a identificação das heurísticas nas mesmas, exemplos e propostas soluções para cada uma das heurísticas em pauta.

Desse modo, as análises serão segmentadas de acordo com os grupos definidos e identificados no Quadro 1, seguindo as características bases de interface (Linha de Comando, Pacote e Interface Gráfica do Usuário).

4.1 Linha de comandos

No campo da bioinformática, não é incomum identificar ferramentas que utilizam única e exclusivamente linha de comando como interface para seus usuários, desse modo, traçar um paralelo entre o custo associado à curva de aprendizado e o uso desse tipo de interface não só é algo plausível como já vem sendo foco de trabalhos do campo, a exemplo do trabalho intitulado “*The missing graphical user interface for genomics*” (SCHATZ, 2010).

No entanto, a discussão com relação ao uso desse tipo de interface apresenta alguns pontos de complexidade. Em comparação a outros campos onde faz-se o uso de soluções computacionais, já que um dos impactos do ritmo acelerado de avanços tecnológicos e redução de custos, na bioinformática, é uma distribuição pouco previsível e pouco estudada entre o conhecimento do uso de linguagens de programação e tecnologias associadas à

implementação de interfaces gráficas para soluções computacionais, tendo pouca relação com a identificação do perfil de experiência do pesquisador, devido a natureza interdisciplinar do campo e o alto grau de especialização necessária para o desempenhar de pesquisas em certos segmentos da bioinformática.

Além dos pontos apresentados, é importante também compreender que a persistência do uso e relevância de ferramentas baseadas em linha de comando pode estar vinculada a complexidade do *workflow* de determinadas pesquisas, pesquisadores e ou tipos de dados, pois, no caso de ferramentas largamente utilizadas no campo com o CIRCOS (KRZYWINSKI et al., 2009), apesar de implementações disponíveis com interface gráfica. Tal cenário pode ser associado a uma necessidade de maior controle sobre a produção de imagens para dados complexos, mantendo certa divisão e relevância entre a distribuição baseada em linha de comando e a distribuição *web* da ferramenta.

Nas coletas desse trabalho, apenas uma ferramenta fez uso desse tipo de interface sendo essa a NG-Circos (CUI et al., 2020), tal uso pode ser justificado pela intenção de aplicar melhorias nos conceitos apresentados pela ferramenta CIRCOS, que justifica o uso de sua interface ao entregar maior controle de suas saídas de dados para seus usuários.

4.2 Pacotes para linguagens de programação

O segmento de ferramentas coletadas formado por pacotes de linguagens de programação é composto como um todo por ferramentas desenvolvidas e distribuídas para a linguagem R.

O uso e desenvolvimento de pacotes para linguagem R apresentam uma série de benefícios e soluções a problemas característicos do campo da bioinformática, sendo esses comumente citados como motivação para o desenvolvimento de novas soluções. Tal qual podemos identificar no trabalho de publicação do pacote *ggtreeExtra* (XU et al., 2021), onde a modularidade trazida por tal distribuição possibilita o uso de soluções em campos que tendem a depender de soluções específicas, dessa forma o uso de pacotes possibilita, de maneira geral, a aplicação de tratamento direcionado para tipos específicos de dados, que são aproveitados em pesquisas de segmentos diferentes, como filogenia, no caso da *ggtreeExtra*, e estudos mais específicos, como no caso da ferramenta *caOmicsV* (ZHANG; MELTZER; DAVIS, 2016), que é direcionada ao tratamento de dados associados a câncer.

Outro aspecto mais complexo é a questão do suporte desses pacotes, já que os mesmos raramente utilizam repositórios comumente utilizados na computação, como os baseados em Git, sendo esse o caso da ferramenta *Rcirc* (SUN; WANG; LI, 2020), que faz uso do GitHub para distribuição de seus códigos, sendo comum o uso de sistemas e repositórios mais especializados no campo da biologia e na distribuição desse tipo de pacote, como o Bioconductor (GENTLEMAN et al., 2004) utilizado pelas ferramentas *karyoploteR* (HU et al., 2014) *OmicCircos* (HU et al., 2014), ou até sistemas de distribuição mais ligados com a linguagem R, como no caso da ferramenta *RCircos* (ZHANG; MELTZER; DAVIS, 2013) e *PACVr* (GRUENSTAEUDL; JENKE, 2020). Tal característica pode afastar usuários não habituados com tais sistemas, como é comum para pesquisadores com maior direcionamento nos campos focados em computação, onde o uso de Git é mais difundido.

Um ponto que vale ressaltar é que, ainda que não seja uma linguagem compreendida como de fácil implementação para interfaces gráficas, existem soluções que agregam os benefícios da linguagem R com interfaces gráfica básicas, como no caso do pacote *Shiny* (“Shiny”, 2022), como exemplo tem-se a ferramenta *shinyChromosome* (YU et al., 2019).

4.3 Graphical User Interface

4.3.1 CIG-P

Como descrito em seu trabalho de publicação, o CIG-P (HOBBS et al., 2014) utiliza de uma interface gráfica implementada com uso da linguagem de programação Java para produzir gráficos circulares e interativos, sendo uma ferramenta direcionada para dados resultantes de proteínas.

Ao longo do trabalho, os autores realizaram a comparação com duas principais ferramentas, sendo essas Cytoscape (SHANNON et al., 2003) e o CIRCOS (KRZYWINSKI et al., 2009). No caso da última, a implicação de uma curva de aprendizado mais custosa acontece não somente baseada da comparação do uso de linha de comando com uma interface gráfica, como também baseada na alta complexidade da estrutura necessária para os dados de entrada da ferramenta CIRCOS, o que, apesar da ausência de comprovação por relatos ou outros métodos que expressariam o entendimento do público dessas ferramentas, pode encontrar suporte na extensa documentação e na própria identificação de relatos similares em outras citações, a exemplo do trabalho de publicação da ferramenta J-Circos (AN et al., 2015), que aponta também para dificuldades de uso provenientes do uso da linguagem de programação Perl no desenvolvimento da ferramenta CIRCOS.

O uso do que pode ser compreendido como diagrama de cordas para a representação de informações é acompanhado apenas de uma hipótese que vincula o uso do deste com uma melhor compressão dos dados apresentados, o que pode indicar a necessidade da realização de trabalhos que verifiquem tal hipótese ou mesmo na ampliação de trabalhos que realizam um agrupamento de hipóteses levantadas por pesquisadores do campo, como no trabalho “*Overview of Genomic Tools for Circular Visualization in the Nextgeneration Genomic Sequencing Era*” (PARVEEN; KHURANA; KUMAR, 2019), já que tais pesquisadores fazem parte do público alvo dessa ferramentas, devido ao perfil acadêmico de suas imagens resultantes.

Analisando a ferramenta em si, pode-se identificar como pontos positivos a facilidade de instalação, tal qual sua disponibilização para múltiplos sistemas operacionais, além da presença de dados de demonstração e fácil utilização desses, sendo essa demonstração observáveis na Figura 2, entrega ao usuário um bom cenário para identificação de funções e do funcionamento como um todo, sem a necessidade de uma compressão total dos múltiplos arquivos necessários para a realização de um resultado inicial ou de teste. A interação com o gráfico por meio de rotação dinâmica e opções de zoom, ainda que não implementada de modo tão dinâmico e similar a *softwares* comuns das suas plataformas nas quais a ferramenta está disponível, provê uma facilidade de leitura e um certo grau de edição, como transformações de tamanho e rotação, limitando-se pela falta de opções de interação e edição para os segmentos do diagrama, como alterações de cor ou adição de rótulo.

No entanto, o uso de arquivos externos para definição de cores e a ausência de edição em tempo real desse recurso, pode implicar na necessidade de atualizações e ou desenvolvimento de novas ferramentas com a mesma finalidade, o que em caso de ausência de reutilização de código poderia também implicar no desuso da padronização de arquivos de entrada identificada nessa ferramenta.

Ainda refletindo sobre aspectos que podem levar a ferramenta ao desuso, a ausência de opções de relato para problemas e de sistemas de trabalho colaborativo como as apresentadas em repositórios como o Github (“Github”, 2022), dificultam a participação

da comunidade interessada na fermenta e exclui a possibilidade de abertura de um canal com seus usuários, dificultando a participação desses pelos desenvolvedores da mesma.

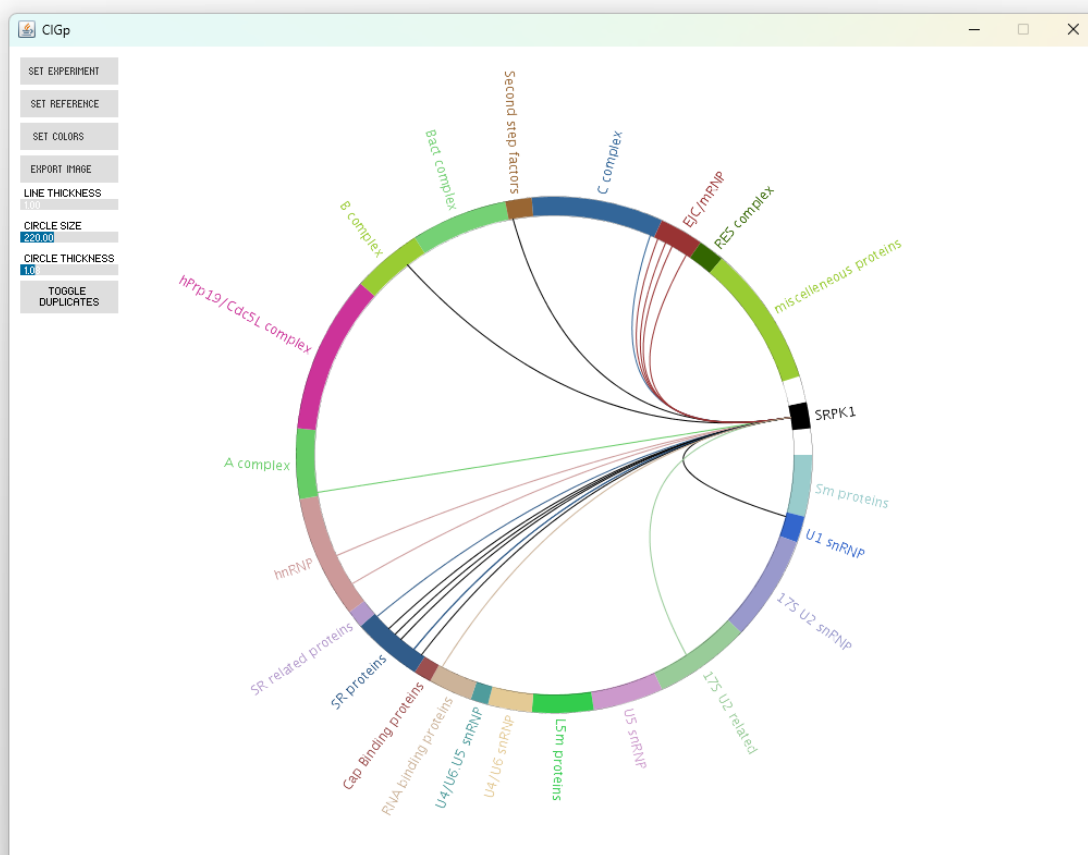


Figura 2 – Captura de tela da ferramenta CIG-P em sua demonstração.

Fonte: Captura própria (2022).

Dentre as heurísticas violadas na ferramenta, a ausência de botões de retorno e barras de navegação dificultam o entendimento do estado e da navegação em si, assim como no caso das telas apresentadas nas Figuras 3 e 4, onde não há mecanismo de retorno entre as duas etapas, forçando o usuário a fechar e reabrir a ferramenta para alteração da opção selecionada no menu identificado na Figura 3, o que implica diretamente nas heurísticas de “Controle e liberdade” e “Reconhecimento em vez de memorização”. Como solução para tal violação, seria necessária a implementação de botões de retorno entre as telas ou uma barra de navegação, essa que também auxiliaria na questão da heurística compreendida pela “Visibilidade de estado no sistema”.



Figura 3 – Captura de tela da ferramenta CIG-P em seu menu de seleção.
Fonte: Captura própria (2022).

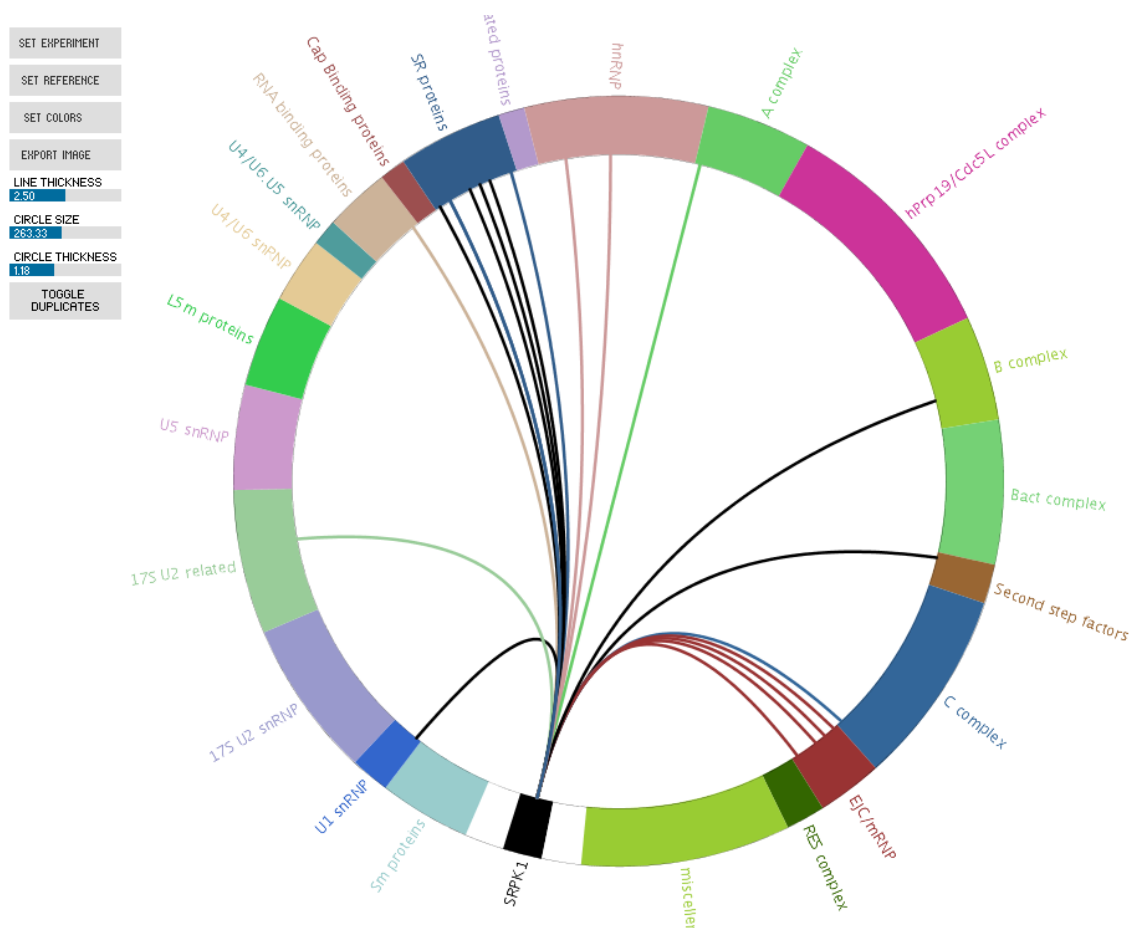


Figura 4 – Captura de tela da ferramenta CIG-P em sua demonstração, com alterações aplicadas sobre o diagrama.
Fonte: Captura própria (2022).

Outra violação a identificada está ligada com as heurísticas de “Reconhecimento em vez de memorização” e “Flexibilidade e eficiência de uso”, pois existem elementos da

interface que apresentam comportamentos diferentes sem qualquer uso de indicações que facilitem o entendimento dessas diferenças, como no caso do botão de “Main Menu”, que pode ser observado na Figura 5, e na similaridade de botões com a opção de *toggle* identificada na Figuras 5 e 3, que neste último caso, por mais que a descrição facilite o uso, seu comportamento apresenta conflitos com a outras interações do gráfico, como a opção de rotação, onde o uso de botões pode resultar em rotações não desejadas. Como solução proposta, deve-se utilizar de ícones para a representação de funções de retorno e alternativas de estados para determinadas funções, facilitando e acelerando o entendimento das diferenças de comportamento e opções disponíveis.

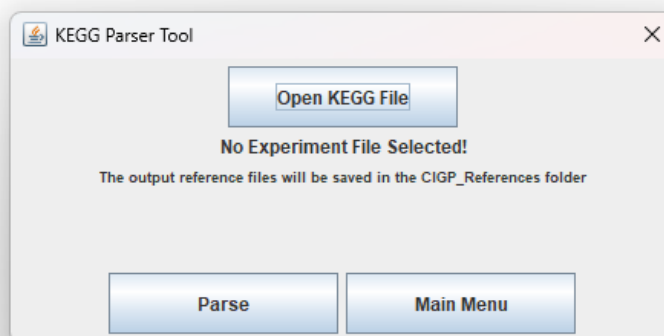


Figura 5 – Captura de tela da ferramenta CIG-P em sua função de *Parser*.

Fonte: Captura própria (2022).

Finalizando a análise apresentada para a ferramenta CIG-P, é evidente também a ausência de mensagens de erro e indicação do estado de funcionamento da mesma, como na tela de seleção apresentada na Figura 6, onde não indicação de erro para entradas inválidas, o que viola as heurísticas “Ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros” e “Visibilidade de estado no sistema”, sendo a implementação de mensagens simples e diretas uma solução válida e simples.

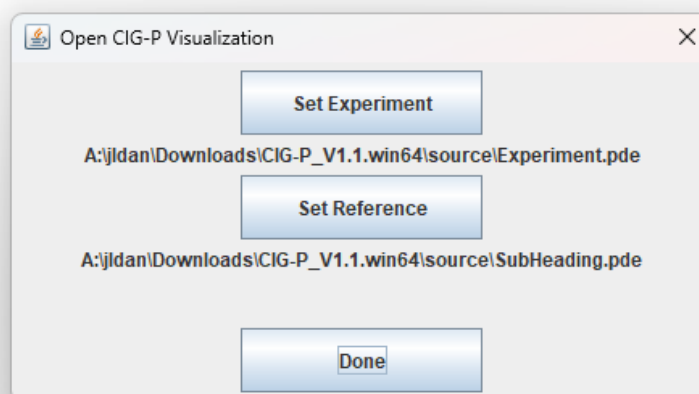


Figura 6 – Captura de tela da ferramenta CIG-P para a seleção de arquivos.

Fonte: Captura própria (2022).

4.3.2 CPGAVAS2

Sendo uma sequência de uma ferramenta previamente desenvolvida e disponibilizada de mesmo modo que sua versão atual, a CPGAVAS2 atua na anotação de realização de análises em dados biológicos, sendo disponibilizada por meio de um site hospedado em um servidor proprietário, porém tendo uma versão baseada em linha de comando que pode ser baixada pelo mesmo site.

A ferramenta CPGAVAS2 (SHI et al., 2019) utiliza de uma estrutura de uso muito similar a um formulário, entregando pouco recurso visual de interação do usuário, apenas botões e seletores de arquivos em sua interface e com a ferramenta propriamente dita, sendo essencialmente um sistema de entrada e saída, sendo a saída observável na Figura 7, o que dificulta muito na curva de aprendizado, porém a disponibilização de demonstrações e arquivos para teste ameniza o impacto negativo na curva de aprendizado da ferramenta.

Retrieve and View the Analysis Results

Please enter the project ID. An ID for a sample annotation run is already filled in the box. Click the submit button directly to view a sample annotation result.

Note

1. If you click on a link and it is redirected to the home page, it means that the corresponding analysis has failed and the corresponding files have not been created correctly. The most likely reason is that your input file(s) have invalid format. Please check the help documents and the sample input files. Make sure your input file(s) meet the requirement. If you have any difficulty, please feel free to contact us.
2. To [download](#) a file, move your cursor to its link, right click your mouse and select "Save As ...".>

1. Gene Identification

1.1 annotation results

1.1.1 GFF3 file.

For details of GFF3 file, please see [here](#). It is recommended that you use [Apollo genome editor](#) to view and edit the annotation.

1.1.2 GenBank file. [Please remember to change the taxon and contact information in this file.](#)

1.1.3 Thumbnail of the schematic genome map.

Figure Legend: Schematic representation of the plastome features. The map contains four rings. From the center going outward, the first circle shows the forward and reverse repeats connected with red and green arcs respectively. The next circle shows the tandem repeats marked with short bars. The third circle shows the microsatellite sequences identified using MISA. The fourth circle is drawn using drawgenemap and shows the gene structure on the plastome. The genes were colored based on their functional categories.

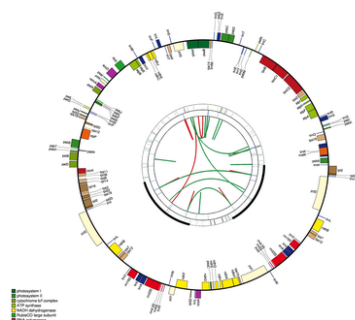


Figura 7 – Captura de tela da ferramenta CPGAVAS2 apresentando resultados.

Fonte: Captura própria (2022).

Do ponto de vista das heurísticas propostas, a ferramenta CPGAVAS2 viola a heurística de “Visibilidade de estado no sistema” e de “Controle e liberdade”, devido a ausência de elementos visuais dinâmico que indiquem o funcionamento da ferramenta durante o tempo de processamento e de finalização desse estado, apresentando também nenhuma descrição do funcionamento de perda ou vinculação de dados processados, onde a ferramenta define um número identificador para o projeto mas não indica a duração do armazenamento do mesmo ou limitações de uso deste identificador. Tal problema poderia

ser solucionado com o uso de elementos de atualização em tempo real ao invés de mensagens fixas, tais mensagens fixas podem ser vistas na Figura 8.

Your job has been submitted and is currently running. It usually takes 20 mins for the annotation to finish. The time to finish the running of extractSeq and AnaDiversity depend on your selection and input data size.

Please keep a note of your project id: 166885275818377, and use it to access your analysis results through <http://47.96.249.172:16019/analyzer/view>

If you have provided an email address, a message will be sent. However, we have seen various problems in receiving the message due to anti-spam policies.

Brief Introduction

The following results can be viewed here:

1. Annotation results
2. Updated annotation results
3. ExtractSeq results
4. AnaDiversity results

Retrieve and View the Analysis Results

Please enter the project ID. An ID for a sample annotation run is already filled in the box. Click the submit button directly to view a sample annotation result.

Figura 8 – Captura de tela da ferramenta CPGAVAS2 durante o processamento de dados.

Fonte: Captura própria (2022).

Finalizando a análise da CPGAVAS2, podemos identificar também a violação da heurística de “Flexibilidade e eficiência de uso”, devido a ausência de ferramentas de edição em tempo real ou sistemas de *preview* para a realização do diagrama, sendo essas características identificáveis nos resultado apresentado na Figura 7, o que seria solucionado com implementação de ferramentas de alterações básica, como modificadores de cores e ferramentas para transformação da imagem.

4.3.3 ClicO FS

A ferramenta ClicOS FS (CHEONG et al., 2015) foi desenvolvido com objetivo de facilitar o uso da ferramenta CIRCOS (KRZYWINSKI et al., 2009), citando a falta de familiaridade de parte do público dessa ferramenta com interface de linha de comando como uma motivação para o desenvolvimento de uma interface gráfica, prezando pela entrega de parte da funcionalidades em um ambiente de menor complexidade para seu usuário final.

No artigo, alguns pontos importantes são definidos com relação a outras soluções que também buscam por entregar a experiência de diagramação da ferramenta CIRCOS por meio de outras soluções computacionais, sendo esses pontos a noção de que essas são, em muitos casos, aplicadas a cenários mais específicos, o que tornam as mesmas menos versáteis, e que em outros casos, não há perspectiva de implantação de interfaces gráfica, como no caso de pacotes para a linguagem R. Ainda no artigo, pouco é descrito com relação ao acesso ao código fonte e, do ponto de vista de IHC, termos como *user friendly* são utilizados de maneira vaga e sem justificativa ou comprovação de eficácia da ferramenta nessa direção.

Levando em consideração a interface implementada, é possível identificar violações de heurísticas como “Consistência e padronização” no caso do uso de conceitos

como *design flat* para certos elementos e aplicação de efeitos de profundidade para outros. Além disso, as inconsistências de interface facilmente confundem o usuário em outros casos, como no caso das setas que são botões, mas não se assemelham a elementos clicáveis como os outros botões, violando as heurísticas de “Reconhecimento em vez de memorização” e “Prevenção de erros”, tais violações podem ser observadas na Figura 9. Como solução proposta, seria necessária uma avaliação dos elementos utilizados, seguido pela aplicação de uma padronização, que poderia, por exemplo, ser proveniente de uma identidade visual desenvolvida para a ferramenta.

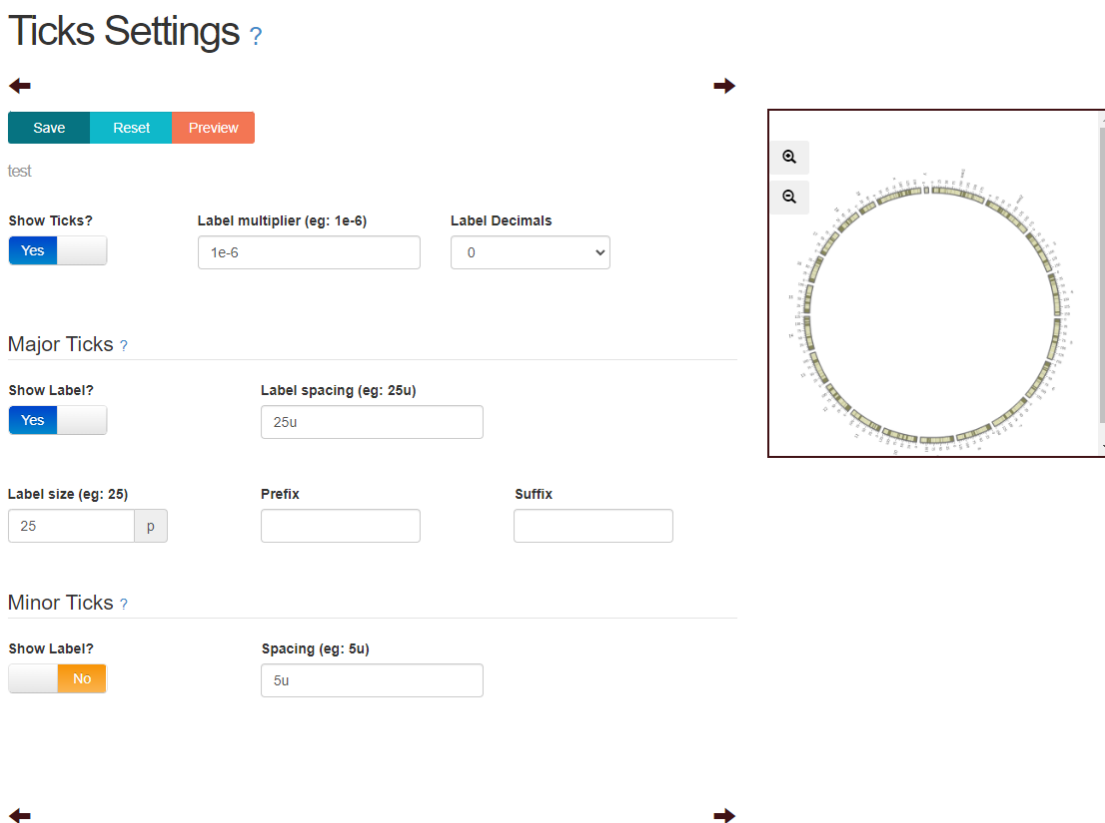


Figura 9 – Captura de tela da ferramenta ClicOS FS.

Fonte: Captura própria (2022).

Por fim, vale ressaltar que tal qual a outras ferramentas analisadas neste trabalho, existem dificuldades de acesso ao código fonte da ferramenta por seus usuários e o uso de servidores proprietários trazem perspectivas negativas do ponto de vista de suporte da comunidade e manutenção no futuro.

4.3.4 FuncTree

FuncTree (UCHIYAMA et al., 2015) é uma ferramenta de visualização para dados biológicos focada em dados provenientes de genômica, porém tendo aplicações para outros segmentos da bioinformática, como metagenômica e transcriptoma. Tal ferramenta busca como diferencial facilitar a compreensão de grandes quantidades de dados, fazendo uso de uma interface gráfica *web* desenvolvida com JavaScript.

Em seu trabalho publicado, é possível identificar uma atenção de seus responsáveis com relação a documentação, já que no próprio trabalho é possível identificar seções que explicam de maneira aprofundada e clara algumas divisões de estruturas de entradas de

dados utilizadas pela ferramenta. Tal característica também é identificada na interface gráfica da ferramenta, onde é facilmente encontrado materiais de suporte e descrições de apoio ao usuário, para situações de ambiguidade e dúvidas, como pode-se observar da Figura 10.



Figura 10 – Captura de tela da ferramenta FuncTree em sua página de ajuda.

Fonte: Captura própria (2022).

A ausência de menção ao código fontes e dificuldade de acesso ao mesmo deve ser vista novamente como um ponto negativo, principalmente pelo uso de um ambiente proprietário e fechado para a distribuição da ferramenta, ou seja sua implantação está associada a um servidor proprietário, o que, assim com citado para outras ferramentas, pode levar a mesma a perder suporte no futuro, devido a dificuldades impostas a participação de terceiros para suportes ou implantações independentes.

Do ponto de vista da avaliação heurística da interface da ferramenta, poucas heurísticas tiveram violações identificadas, dentre essa, pode-se citar a heurística de “Controle e liberdade”, pois a ferramenta não apresenta mecanismos de retorno para estados de funcionamento em casos de erros, como na inserção de arquivos inválidos apresentado da Figura 11. Ainda nesse cenário é possível identificar que o usuário perde compressão do estado funcionamento da aplicação, que no caso de não funcionamento permanece indicando o carregamento de uma ação inexistente, o que resulta na violação da heurística de “Visibilidade de estado no sistema”. Ao analisar e propor uma solução para esse cenário, foi possível identificar a violação da heurística de “Prevenção de erros”, já que um tratamento de dados e filtros na inserção desse solucionaria o problemas da violações citadas de maneira total, ainda que situações como do carregamento infinito possam vir a aparecer em outros cenários, o que reforça a necessidade de mensagens e ícones mais dinâmicos para demonstrar o funcionamento em si das análises que rodam por trás da interface gráfica

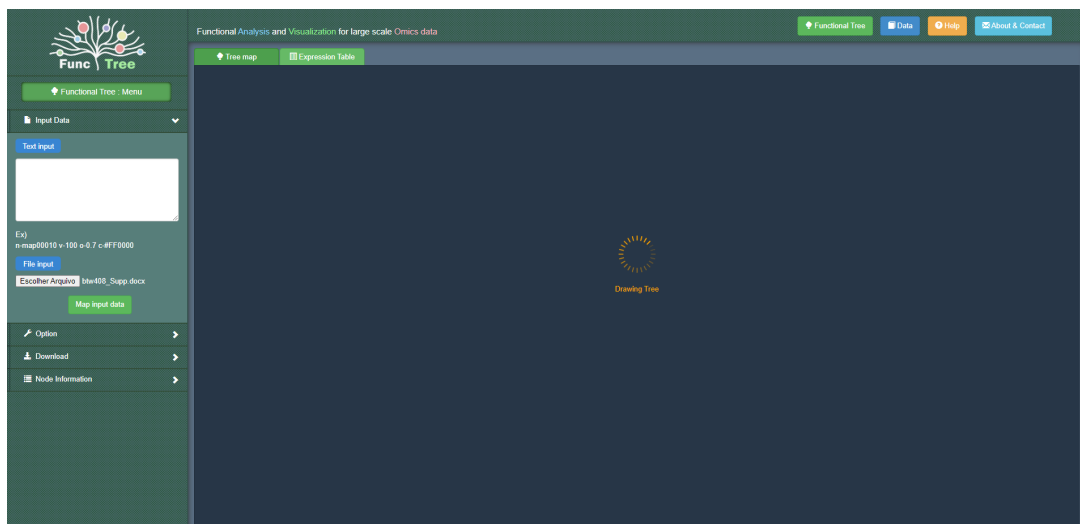


Figura 11 – Captura de tela da ferramenta FuncTree após inserção inválida de dados.

Fonte: Captura própria (2022).

4.3.5 OGDRAW

A ferramenta OrganellarGenomeDRAW (LOHSE et al., 2013) ou OGDRAW, tem como função a realização de imagens para expressar o mapeamento genomas, fazendo uso de bancos de dados externo para facilitar e tornar o processo de gerar imagens dessa natureza mais rápidos e simples, o que pode justificar o uso de interface gráfica em detrimento da linha de comando.

Analisando a ferramenta, o fluxo de funcionamento faz uso de poucos objetos gráficos dinâmicos, podendo ser resumida a seleção de dados, opções de plotagem e a imagem resultante, dessa forma, é possível dizer que o uso de interface gráfica é pouco explorado e traz um modelo similar a um formulário, com uso de *checkboxes*, seletor de arquivos e alguns outros elementos estáticos, como pode-se observar na Figura 12.

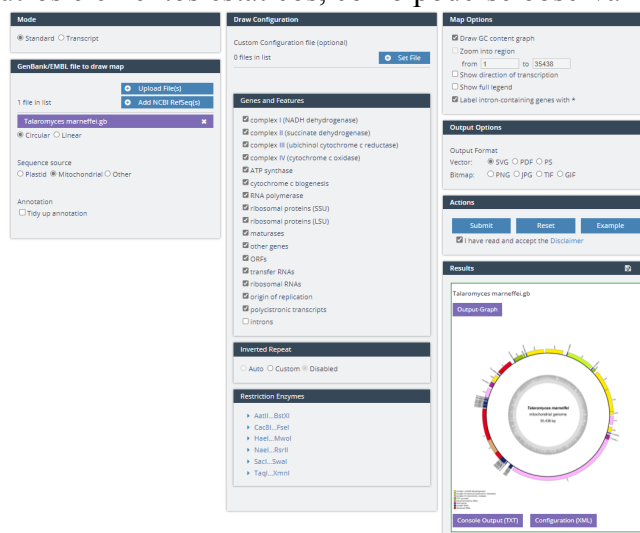


Figura 12 – Captura de tela da ferramenta OGDRAW após a plotagem de uma imagem.

Fonte: Captura própria (2022).

A partir do entendimento do fluxo de funcionamento da ferramenta, é possível identificar a violação da heurística “Flexibilidade e eficiência de uso”, já que não há uso de elementos que encurtem tarefas em relação a uma implementação com uso de linha de comando como principal interface. Para solucionar essa questão, é necessário implementar ferramentas de edição e visualização em tempo real, o que também atuaria diretamente na violação da heurística de “Visibilidade de estado no sistema”, pois a ferramenta OGDRAW não apresenta elementos de transparência para a indicação de estado de funcionamento de suas plotagens.

Outra violação de uma heurística é relacionada a “Ajuda e documentação”, já que a ferramenta não apresenta de maneira clara um método de acesso a documentações de apoio, o que seria solucionado com uma simples página dedicada a tal documentação ou a adição de botões para descrições de funcionamento e de opções diretamente na interface.

Por fim, vale ressaltar que, seria um ponto positivo a disponibilização do código fonte por meio de repositórios públicos, ainda que a presença da mesma em um ambiente dedicado para várias ferramentas indique uma maior segurança em termos de suporte, pois tal disponibilização permitiria maior facilidade para reportar problemas e para adicionar possíveis melhorias, além de aumentar a independência da ferramenta em relação a implantação em servidores de terceiros.

5. DISCUSSÃO

É possível identificar que, dentre as ferramentas coletadas, ainda que a genômica comparativa seja um dos principais segmentos do uso de visualização de dados circulares no contexto da bioinformática, diversas finalidades e segmentos de pesquisas, desse contexto, encontram na visualização circular um método de produzir imagens de fácil entendimento para grandes agrupamentos de dados biológicos, como em ferramentas previamente citadas focadas em segmentos como da filogenia ou até mesmo doenças específicas, portanto, tal complexidade inerente a natureza da diversidade das pesquisas da área, levam a produção de ferramentas específicas para cada cenário, justificando a grande quantidade de ferramentas produzidas com aspectos similares. Para mitigar essa questão, deve-se prezar por soluções que permitam uma maior modularização de ferramentas, como no caso de desenvolvimento e distribuição de pacotes, tal qual os pacotes desenvolvidos para linguagem R, sem deixar de lado questões como uso e desenvolvimento de interfaces que levam em consideração os seus diversos usuários.

Outro aspecto, que pode ter grande impacto na quantidade de ferramentas produzidas, é a questão da padronização de entradas para as ferramentas, pois, devido a complexidade e alta quantidade de dados, o cenário atual da bioinformática apresenta pouca padronização e estudos direcionados para as entradas recebidas pelas ferramentas, sendo prática comum definições arbitrárias e pouco relacionadas para ferramentas com objetivos comuns e atuação em segmentos similares da bioinformática.

As ferramentas baseadas em distribuição *web* também apresentam um problema comum, a vinculação e dependência de servidores proprietários, já que não é incomum encontrar dificuldades de acesso, na procura por um código fonte e ou na procura por documentos e materiais de apoio para tais ferramentas, inviabilizando implantação em servidores independentes e ou suporte de terceiros, o que resulta na dependência dos servidores disponibilizados pelos seus desenvolvedores, que por muitas vezes acaba por perder suporte e, em alguns casos, levam as ferramentas a estados completamente inacessível ao longo do tempo, graças em parte pela dificuldade e custo elevado para manter sistemas como esses.

Nesse aspecto de servidores proprietários, existem algumas soluções que buscam atuar nessas questões, uma delas é a disseminação de ambientes colaborativos ou plataformas, como as largamente utilizadas Galaxy (AFGAN et al., 2018), que buscam por agrupar múltiplas ferramentas em um único ambiente, facilitando o trabalho de manter o mesmo e o próprio *workflow* associado a um conjunto de ferramentas ou dados dos pesquisadores, porém tal iniciativa não resolve o problema como um todo, o que depende de uma disponibilização do código fonte para implementações em ambientes de dados sensíveis por exemplo, onde o *upload* de dados biológicos não é viável, ou para usos de determinados trechos de códigos em outras aplicações, que para seu funcionamento pleno e independente necessitam comumente de alterações e implementações realizadas diretamente a nível de código.

Refletindo sobre o uso de linguagens de programação, metodologias de desenvolvimento e seleção de plataformas, é difícil traçar paralelos entre as ferramentas, porém não é raro identificar o uso de linguagens sem suporte atual, como o uso de ActionScript, o que inviabiliza a análise no caso da ferramenta SynTView (LECHAT; SOUCHE; MOSZER, 2013) e o uso em si da ferramenta, ou o uso de linguagens que vêm perdendo espaço entre a comunidade de programação como Perl, o que, ainda que não implique em problemas diretos para ferramentas escritas em linguagens dessa natureza, acaba por dificultar o desenvolvimento, principalmente para programadores com pouca experiência e advindos de outros campos, pela falta de suporte da comunidade em fóruns de perguntas e falta de programadores com interesse em apoiar tais projetos.

Com relação aos testes de usabilidade e identificação das heurísticas, utilizando o conjunto de heurísticas proposto, é possível afirmar que pouco cuidado é levado em determinados aspectos pelas ferramentas que apresentam implementações de interfaces gráficas, sendo identificados ausências de fatores e podem ser entendidos como críticos, como no caso de ausência de mensagens de erros e respostas indicadoras do estado de funcionamento da aplicação, demonstrando-se que é necessária um cuidado com o básico em termos de IHC Nesse aspecto é importante que sejam realizadas pesquisas mais direcionadas, a fim de produzir sistemas e métodos de avaliação que estejam de acordo com a área e que sejam acessíveis para tais profissionais, sendo necessário um esforço colaborativo para o entendimento de tais necessidades em busca de um equilíbrio entre os níveis de conhecimento e as diversas áreas de atuação que são naturais para campos interdisciplinares, como no caso da bioinformática.

A fim de mapear quais aspectos foram mais compreendidos como importantes, fora realizado um gráfico para compreensão de quais heurísticas foram mais identificados como violadas, a fim de servir com possível guia para futuras avaliações, retrabalho de interface e desenvolvimento de novas ferramentas, tal gráfico pode ser observado na Figura 13. Além disso, para ferramentas que apresentam interface gráfica, foi realizado uma análise quantitativa da frequência do uso de termos comumente associados com o campo de IHC, a fim trazer *insights* sobre a importância e compressão desse termos, no entanto termos comuns de grande importância para aspecto de IHC, como “Accessibility”, “User Experience”, “Usability” e “Communicability” não foram citados em nenhum dos trabalhos de publicação das ferramentas onde a avaliação heurística foi aplicada.

Frequência da identificação de violação de heurística

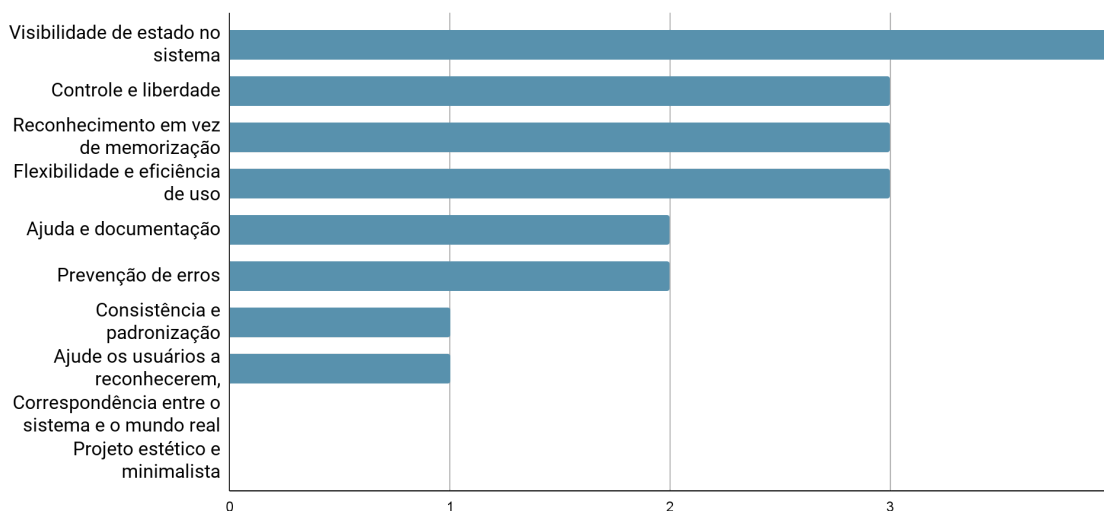


Figura 13 – Gráfico de frequência de identificação de violação de heurística.

Fonte: Elaboração própria (2022).

6. CONCLUSÃO

Dentre os resultados identificados, pode-se afirmar que, no campo de bioinformática, os problemas de interfaces gráficas implementadas com foco em visualização circular estão diretamente vinculados com questões associadas às heurísticas mais frequentemente identificados como violadas, portanto a produção do gráfico, observável na Figura 13, permite a desenvolvedores *insights* para pontos de melhorias para as ferramentas aqui analisadas, resultando também em como um guia para pesquisas de perspectivas de IHC do ponto de vista do desenvolvimento de novas ferramentas, que devem guiar-se por trabalhos que focam justamente em propostas de soluções para tais violações dessas heurísticas.

Além disso, partir da produção desse trabalho, fica evidente que, para campos de conhecimento dependentes e participantes de paradigmas computacionais, é importante considerar aspectos da computação que beneficiam tal campo, utilizando sempre como base o trabalho realizado previamente para levantar novos debates e consequentemente identificar os pontos de melhorias comuns entre os campos. No entanto, vale ressaltar que, não apenas por uma alta complexidade e uma grande quantidade de dados inerente ao campo, é de grande importância compreender quem são os usuários que fazem uso de tais ferramentas, já que, nesse aspecto, o campo da bioinformática traz desafios e perspectivas únicas que foram citadas ao longo deste trabalho, reforçando a ideia de que não há como definir e formular métodos ou definições sistemáticas e universais de propostas de melhorias para o campo, gerando ainda mais desafios para disseminação e compartilhamento de estudos que abordem metodologia de desenvolvimento, IHC e outros paradigmas da computação e da biologia nesse cenário.

Para trabalhos futuros, propostas e pesquisas de aspectos de avaliação associados a usabilidade de *softwares* para o contexto de bioinformática são imprescindíveis para que haja um avanço palpável na qualidade do *software* desenvolvido nesse campo, portanto é de grande importância o desenvolvimento de pesquisas com propostas para tais critérios. Outro aspecto similar que pode ser explorado, em futuras pesquisas, seria a identificação das metodologias comuns para o desenvolvimento em bioinformática suas relações com as

metodologias de *softwares* estudadas e aplicadas e outros campos, traçando paralelos e comparações com o estado da artes em engenharia de *software*, o que permitiria, entre outros, a comparação direta entre metodologias ágeis e mais recentes com o que seria identificado como posto em prática pela comunidade de bioinformática. Por fim, as limitações deste trabalho justificam a produção de novas avaliações heurísticas sobre as ferramentas analisadas, a fim de expandir nos aspectos de quantidade de avaliadores e no aspecto de maior inclusão de ferramentas para um escopo maior, o que resultaria em uma maior qualidade de análise ferramentas e das perspectivas aplicadas sobre as avaliações em si, além do aumento quantitativo.

REFERÊNCIAS BIBLIOGRÁFICAS

- AFGAN, E. et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. **Nucleic Acids Research**, v. 46, n. W1, p. W537–W544, 2 jul. 2018.
- AN, J. et al. J-Circos: an interactive Circos plotter. **Bioinformatics (Oxford, England)**, v. 31, n. 9, p. 1463–1465, 1 maio 2015.
- Biological Data Visualization Market Size & Share Report, 2030**. Disponível em: <<https://www.grandviewresearch.com/industry-analysis/biological-data-visualization-market>>. Acesso em: 15 nov. 2022.
- CHEONG, W.-H. et al. ClicO FS: an interactive web-based service of Circos. **Bioinformatics (Oxford, England)**, v. 31, n. 22, p. 3685–3687, 15 nov. 2015.
- CUI, Y. et al. NG-Circos: next-generation Circos for data visualization and interpretation. **NAR genomics and bioinformatics**, v. 2, n. 3, p. lqaa069, set. 2020.
- FLEISCHMANN, R. D. et al. Whole-Genome Random Sequencing and Assembly of *Haemophilus influenzae* Rd. **Science**, v. 269, n. 5223, p. 496–512, 28 jul. 1995.
- GENTLEMAN, R. C. et al. Bioconductor: open software development for computational biology and bioinformatics. **Genome Biology**, v. 5, n. 10, p. R80, 15 set. 2004.
- GitHub: Let's build from here**. Disponível em: <<https://github.com/>>. Acesso em: 18 nov. 2022.
- GRUENSTAEUDL, M.; JENKE, N. PACVr: plastome assembly coverage visualization in R. **BMC bioinformatics**, v. 21, n. 1, p. 207, 24 maio 2020.
- Guide to Application of Circos in Comparative Genomics and Visualizing Sequence Similarity and Conservation // CIRCOS Circular Genome Data Visualization**. Disponível em: <<http://circos.ca/guide/genomic/>>. Acesso em: 18 nov. 2022.
- HOBBS, C. K. et al. CIG-P: Circular Interaction Graph for Proteomics. **BMC Bioinformatics**, v. 15, n. 1, p. 344, 31 out. 2014.
- HU, Y. et al. OmicCircos: A Simple-to-Use R Package for the Circular Visualization of Multidimensional Omics Data. **Cancer Informatics**, v. 13, p. 13–20, 2014.
- KRZYWINSKI, M. et al. Circos: an information aesthetic for comparative genomics. **Genome Research**, v. 19, n. 9, p. 1639–1645, set. 2009.
- LECHAT, P.; SOUCHE, E.; MOSZER, I. SynTView - an interactive multi-view genome browser for next-generation comparative microorganism genomics. **BMC bioinformatics**, v. 14, p. 277, 22 set. 2013.
- LOHSE, M. et al. OrganellarGenomeDRAW--a suite of tools for generating physical maps of plastid and mitochondrial genomes and visualizing expression data sets. **Nucleic Acids Research**, v. 41, n. Web Server issue, p. W575-581, jul. 2013.
- MOLICH, R.; NIELSEN, J. Improving a human-computer dialogue. **Communications of the ACM**, v. 33, n. 3, p. 338–348, 1 mar. 1990.

- National Institutes of Health (NIH)**. Disponível em: <<https://www.nih.gov/>>. Acesso em: 27 dez. 2022.
- PARVEEN, A.; KHURANA, S.; KUMAR, A. Overview of Genomic Tools for Circular Visualization in the Next-generation Genomic Sequencing Era. **Current Genomics**, v. 20, n. 2, p. 90–99, fev. 2019.
- PubMed**. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/>>. Acesso em: 27 dez. 2022.
- SCHATZ, M. C. The missing graphical user interface for genomics. **Genome Biology**, v. 11, n. 8, p. 128, 2010.
- SHANNON, P. et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. **Genome Research**, v. 13, n. 11, p. 2498–2504, nov. 2003.
- SHI, L. et al. CPGAVAS2, an integrated plastome sequence annotator and analyzer. **Nucleic Acids Research**, v. 47, n. W1, p. W65–W73, 2 jul. 2019.
- Shiny**. Disponível em: <<https://shiny.rstudio.com/>>. Acesso em: 18 nov. 2022.
- SUN, P.; WANG, H.; LI, G. Rcirc: An R Package for circRNA Analyses and Visualization. **Frontiers in Genetics**, v. 11, p. 548, 2020.
- UCHIYAMA, T. et al. FuncTree: Functional Analysis and Visualization for Large-Scale Omics Data. **PloS One**, v. 10, n. 5, p. e0126967, 2015.
- We make healthcare run better**. Disponível em: <<https://www.medline.com/>>. Acesso em: 27 dez. 2022.
- XU, S. et al. ggtreeExtra: Compact Visualization of Richly Annotated Phylogenetic Data. **Molecular Biology and Evolution**, v. 38, n. 9, p. 4039–4042, 23 ago. 2021.
- YU, Y. et al. shinyChromosome: An R/Shiny Application for Interactive Creation of Non-circular Plots of Whole Genomes. **Genomics, Proteomics & Bioinformatics**, v. 17, n. 5, p. 535–539, out. 2019.
- ZHANG, H.; MELTZER, P.; DAVIS, S. RCircos: an R package for Circos 2D track plots. **BMC bioinformatics**, v. 14, p. 244, 10 ago. 2013.
- ZHANG, H.; MELTZER, P. S.; DAVIS, S. R. caOmicsV: an R package for visualizing multidimensional cancer genomic data. **BMC bioinformatics**, v. 17, p. 141, 22 mar. 2016.