PATO: um ambiente integrado com interface gráfica para a curadoria de dados de sequências biológicas

Liliane Santana Oliveira

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Pós-Graduação em Ciência da Computação
Orientador: Prof. Dr. Alan Mitchell Durham

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro do CNPq

São Paulo, Agosto de 2013

PATO: um ambiente integrado com interface gráfica para a curadoria de dados de sequências biológicas

Este exemplar corresponde à redação
final da dissertação devidamente corrigida
e defendida por Liliane Santana Oliveira
e aprovada pela Comissão Julgadora.

Banca examinadora:

- Profº Dr Alan Mitchell Durham - IME-USP
- Profº Dr André Fugita - IME-USP
- Profº Dr Vasco Ariston de Carvalho Azevedo - UFMG

Agradecimentos

Agradeço primeiramente a Deus, por me dar coragem durante esses três anos de muito trabalho. Aos meus pais, Maria José e Geraldo, pelo apoio em todas as áreas da minha vida. Eu não seria a pessoa que sou sem vocês e não teria chegado até aqui. Aos meus irmãos, Lidiane e João Geraldo, pelo carinho e apoio durante toda a vida. Aos meus orientadores, Alan Mitchell e Arthur Gruber, pelos conhecimentos passados, paciência e por todo esforço que fizeram por mim durante esse tempo. À minha segunda família (vó Alaide, tias Lourdes e Ana e primas Maria José e Rosane), tios e minha madrinha, Rita, pelo carinho e apoio em todas as situações. A um grande amigo feito no caminho, Alexandre Rossi, por não medir esfoços para me ajudar, e a todos os amigos do grupo de pesquisa. À equipe do Instituto de pesquisa Albert Einstein, pela compreensão da ausência em alguns momentos e pelas boas risadas nos momentos de tensão (em especial Eliane Antonioli, Natália Torres, Patrícia Severino, Marta Jardim, Andrea Vieira e Luiz Sardinha). Aos amigos de longa data e aos encontrados em São Paulo, em especial Ulysses Oliveira, Kátia Ribeiro, Leisa Cunha e Juliana Lopes, muito obrigada pelos momentos de distração e desabafo. Sem vocês eu teria enlouquecido! Enfim, muito obrigada a todos que direta ou indiretamente contribuiram para que eu chegasse aqui.

Resumo

OLIVEIRA, L. S. **PATO: um ambiente integrado com interface gráfica para a curadoria de dados de sequências biológicas**.

A evolução das tecnologias de sequenciamento de DNA tem permitido a elucidação da sequência genômica de um número cada vez maior de organismos. Contudo, a obtenção da sequência nucleotídica do genoma é apenas a primeira etapa no estudo dos organismos. O processo de anotação consiste na identificação as diferentes regiões de interesse no genoma e suas funcionalidades. Várias ferramentas computacionais foram desenvolvidas para auxiliar o processo de anotação, porém nenhuma delas permite ao usuário selecionar sequências, processá-las de forma a encontrar evidências a respeito das regiões genômicas, como predição gênica e de domínios protéicos, analisá-las graficamente e adicionar informações a respeito de suas regiões em um mesmo ambiente. Assim, o objetivo desse projeto foi o desenvolvimento de uma plataforma gráfica para a anotação genômica que permite ao usuário realizar as tarefas necessárias para o processo de anotação em uma única ferramenta integrada a um banco de dados. A idéia é proporcionar ao usuário liberdade para trabalhar com o seu conjunto de dados, possibilitando a seleção de sequências para análise, construção dos *pipelines* processamento das mesmas e análise dos resultados encontrados a partir de visualizador que permite ao usuário adicionar informações às regiões e fazer a curadoria das sequências. A ferramenta resultante é facilmente extensível, permitindo o acoplamento modular de novas funcionalidades de anotação e sua estrutura permite ao usuário trabalhar tanto com projetos de sequências expressas como anotação de genomas.

Palavras-chave: anotação, ambiente gráfico, processamento.

Abstract

OLIVEIRA, L. S. **PATO: an integrated enviroment with GUI to data curation of biological sequences**.

The evolution of the technologies of DNA sequencing has permitted the elucidation of genomic sequence of an increasing number of organisms. Though, the obtainment of the genome nucleotide sequence is only the first step in the study of organisms. The annotation process consists in the identification of different regions of interest on the genome and their features. Several computational tools were developed to support the annotation process, however none allow the user to select sequences, process them, analyze them graphically and add information about its regions in the same surrounding. Thus, the aim of this project was to develop a graphic platform to genome annotation that allows the user to realize your tasks required from the annotation process in a single tool integrated to a database. The idea is to provide from the user liberty to work with your dataset, enabling the selection of sequences for analyze, pipeline construction, processing them and analyze of results from the viewer that allows the user to add information in the regions and to do the trusteeship of sequences. The resulting tool is easily extensible; allowing the engagement modular of new functionalities of annotation and its structure allows the user works both projects of expressed sequences and with genome annotation.
Palavras-chave: annotation, graphic platform, processing.

Sumário

Lista de Figuras

Capítulo 1

Introdução

A vida depende da capacidade da célula de armazenar, recuperar e traduzir informações genéticas requeridas para manter um organismo vivo, que são passadas de célula para célula no processo de divisão celular [AJL+08]. Estas informações estão contidas em uma molécula conhecida como DNA ou *deoxyribonucleic acid* (em português ácido dexorribonucléico - ADN), que consiste de duas longas cadeias compostas por quatro tipos de nucleotídeos[1]. O DNA humano tem por volta de três bilhões de bases, sendo que cerca de 99% delas são iguais para todos os humanos, e pode ser encontrado no núcleo celular (em sua maior parte) e nas mitocôndrias. A disposição dessas bases determina a informação disponível para a construção e manutenção de um organismo. O processo de descoberta da ordem dos nucleotídeos em um segmento de DNA é definido como sequenciamento, e é realizado por equipamentos denominados sequenciadores.

A evolução das tecnologias de sequencimento foi capaz de gerar plataformas com custos de sequenciamento cada vez menores e capazes de gerar informações a respeito de milhões de pares de bases em uma única execução. O aumento da capacidade de sequenciamento tem permitido o estudo de uma grande quantidade de genomas antes desconhecidos, possibilitando, inclusive, o estudo dos metagenomas[2].

Porém o sequenciamento é apenas a primeira etapa do processo, uma vez que é preciso identificar os elementos da sequência, como genes codificantes de proteína e de *RNA* não codificante, e suas respectivas funções. Esse processo é definido como anotação e consiste em

[1] Compostos ricos em energia que auxiliam os processos metabólicos.

[2] Um metagenoma é o genoma da microbiota total encontrado em um determinado habitat, que pode ser um ambiente qualquer, como o chão ou o estômago humano.

atribuir significado biológico às sequências brutas resultantes de algum processo de sequenciamento, de forma a auxiliar biólogos e pesquisadores [Lin01]. Para maior clareza, dividimos esse processo em três etapas: identificação dos elementos da sequência, busca funcional desses elementos e síntese das informações a partir dos resultados encontrados a fim de gerar uma conclusão a respeito das regiões das sequência. Várias ferramentas realizam tarefas que auxiliam o processo de anotação, podendo executar tarefas isoladas relacionadas a algumas das três fases, como predição de gene ou análise funcional de uma região, ou fazer parte de um sistema integrado que realiza várias tarefas de anotação. A nossa pesquisa bibliográfica encontrou uma ferramenta que realiza seleção, processamento e análise de sequências em uma mesma plataforma, o *GenDB* [MGM+03]. Porém a ferramenta somente funciona plenamente no ambiente construído pelos desenvolvedores da plataforma, o que uma limitação quando se trabalha com sequências que não podem ser submetidas a terceiros.

Assim, o objetivo desse projeto foi o desenvolvimento de uma plataforma gráfica para a anotação genômica que permite ao usuário realizar as tarefas necessárias para o processo de anotação em uma única ferramenta. Na plataforma, o processamento das sequências é realizado a partir da execução de fluxos de processamento definidos pelo usuário sendo que os resultados encontrados são integrados à sequência e exibidos graficamente. A ferramenta permite ao usuário analisar os resultados, adicionar informações e realizar novos processamentos. Podemos destacar uma característica particular da ferramenta, que é a possibilidade do usuário selecionar um subconjunto de sequências para análise, ao contrário das ferramentas existentes, nas quais o usuário trabalha com todo o conjunto de entrada (nenhuma das ferramentas existentes possui um mecanismo para a seleção de sequências). Como critério de seleção, o usuário pode utilizar tanto resultados de processamentos anteriores (como pontuações ou propriedades atribuídas aos resultados encontrados) como propriedades das sequências (como organismo a que pertencem). As sequências selecionadas podem seguir para processamento ou para análise visual. Para fins de padronização, as informações serão adicionadas às regiões das sequências a partir de termos de vocabulários controlados e o usuário pode anexar informações textuais através de comentários. A plataforma está integrada a uma base de dados, permitindo que todas as ações realizadas, bem como resultados encontrados, sejam armazenados e recuperados sempre que necessário.

O desenvolvimento da ferramenta foi realizado sob a co-orientação informal do professor Dr. Arthur Gruber, pesquisador com extensa experiência em anotação e co-desenvolvedor do *EGene*, visando garantir que a funcionalidade e a interface se aproximassem o máximo das necessidades do público alvo, biólogos envolvidos no processo de anotação.

Este documento está organizado da seguinte forma: o capítulo seguinte descreve o problema da anotação e alguns conceitos necessários para o entendimento do trabalho, bem como faz a revisão das ferramentas existentes na literatura. O capítulo três trata da plataforma em si, descrevendo a sua arquitetura e a modelagem utilizada. Por fim, o último capítulo traz as conclusões e os trabalhos futuros.

Capítulo 2

Sistemas de apoio à anotação

O processo de anotação consiste em descrever as características presentes em sequências biológicas. Para cada característica existe um par de coordenadas de localização e as descrições podem ser de diferentes naturezas, incluindo a identificação de genes e suas proteínas, funções moleculares e promotores. No presente trabalho, classificamos esse processo em etapas: identificação das características (*features*) presentes na sequência, identificação das funções dessas *features* e síntese das informações a partir dos resultados encontrados, gerando uma conclusão que descreva detalhadamente tais *features*.

A partir da revisão da literatura é possível encontrar ferramentas que realizam tarefas que auxiliam o processo de anotação. Tais ferramentas podem executar tarefas isoladas relacionadas a algumas das três fases, como predição de gene ou busca de similaridade, ou fazer parte de um sistema que integra um conjunto de recursos e desempenha tarefas de anotação.

Como resultado do processo de anotação são geradas informações a respeito das regiões da sequência. A forma mais simples de gerá-las é a partir de texto livre. Esse formato não possui padronização de termos, permitindo que as regiões sejam descritas de acordo com o especialista que realiza a análise. Essa liberdade possibita que regiões com mesmas características sejam descritas com termos distintos, gerando ambiguidade e dificultando a mineração dos dados. Com o objetivo de padronização dos termos de anotação, no final da década de 1990 começaram a ser desenvolvidos vocabulários controlados e ontologias, como *GO (Gene Ontology) [Con00]* e *SO (Sequence Ontology) [ELM+05]*.

A evolução das tecnologias de sequenciamento aumentou consideravelmente o volume de

genomas sequenciados. Esse aumento fez surgir a necessidade da utilização de bancos de dados para armazenamento de tais informações, resultando no desenvolvimento de vários bancos de dados para armazenamento dessas informações.

Esse capítulo descreve o processo de anotação. Para seu entendimento completo, alguns conceitos biológicos, como o dogma central da biologia, foram adicionados. Este capítulo descreve ainda as principais ferramentas utilizadas no processo de anotação de sequências

2.1 Conceitos biológicos

Um gene é uma região da sequência genômica que corresponde a uma unidade de hereditariedade, a qual se associam regiões regulatórias, transcritas e/ou outras regiões funcionais da sequência [Pea06]. O processo de síntese de proteína é parte do dogma central da biologia molecular, como ilustrado na figura 2.1. A primeira etapa do processo de síntese de proteína é denominado transcrição e consiste na síntese de um RNA a partir do DNA. Tal processo ocorre a partir da separação da dupla hélice do DNA, de forma que suas fitas fiquem expostas e uma delas sirva como molde para a síntese da molécula de RNA. Dessa forma, a sequência nucleotídica do RNA é determinada pela complementariedade entre os nucleotídeos de entrada e a fita molde e o RNA sintetizado é denominado de transcrito. A figura 2.2 ilustra o processo de trancrição.

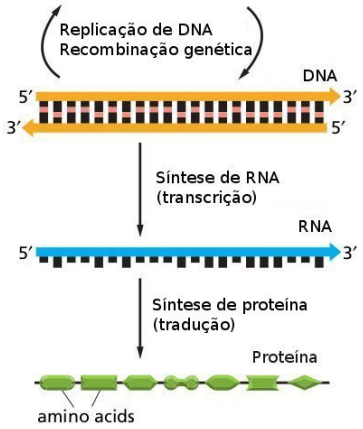


Figura 2.1: A figura ilustra o dogma central da biologia molecular. Figura extraída e modificada de [AJL+08].

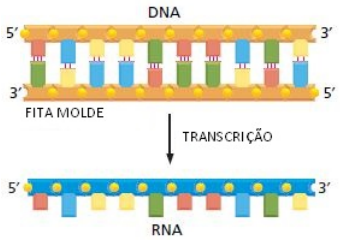


Figura 2.2: Transcrição de uma molécula de *RNA*. Figura extraída e modificada de [AJL+08].

A figura 2.3 ilustra a estrutura primária dos genes eucarióticos codificante de proteína. Nesse caso, o gene pode formado por segmentos de sequências expressas, denominadas *éxons*, intervalados por segmentos não codificadores, denominadas *íntrons* [Kas11]. Ambos *íntrons* e *éxons* são transcritos em RNA, porém os *íntrons* são removidos do RNA através do processo definido como *splicing* de RNA. O RNA resultante do splicing é um RNA mensageiro (ou mRNA) formado apenas pelos *éxons*. A parte que antecede a região codificante é denominada de região promotora, ou sítio de transcrição, e pode ser conservada, rica em bases *A* ou *T* (conhecida por *TATA-box*). O gene eucarioto possui ainda regiões não codificantes em suas extremidades, conhecidas como *UTRs (untranslated regions)*. Os genes de organismos procariotos são formados apenas por um trecho contínuo de DNA, que é diretamente transcrito em mRNA, ou seja, o processo de *splicing* não acontece nos genes dos organismos procariotos.

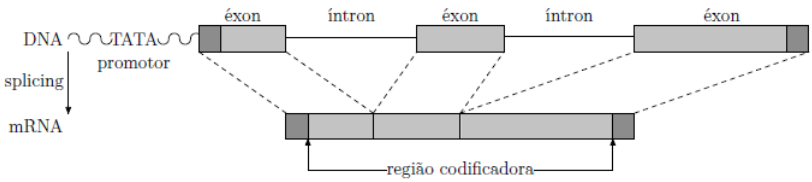


Figura 2.3: Estrutura de um gene codificador de proteína de um organismo eucarioto. Figura extraída e modificada de [Kas11].

Uma vez que o *mRNA* foi produzido e processado de forma a eliminar os *íntrons*, as informações presentes em sua sequência de nucleótidos são utilizadas para sintetizar uma proteína. A conversão da informação do RNA em proteína é conhecida por tradução. Neste processo, os nucleotídeos do *mRNA* são lidos três a três de forma consecutiva, e cada trinca de nucleotídeos (definido como um códon) corresponde a um aminoácido específico. A tradução de um códon em um aminoácido depende de uma molécula adaptadora, denominada de tRNA (RNA

transportador), que reconhece a tradução e liga o aminoácido ao códon. A figura 2.4 ilustra a tabela de uso de códon padrão, a partir da qual é possível visualizar a relação dos aminoácidos existentes com seus respectivos códons.

GCA GCC GCG GCU	AGA AGG CGA CGC CGG CGU	GAC GAU	AAC AAU	UGC UGU	GAA GAG	CAA CAG	GGA GGC GGG GGU	CAC CAU	AUA AUC AUU	UUA UUG CUA CUC CUG CUU	AAA AAG	AUG	UUC UUU	CCA CCC CCG CCU	AGC AGU UCA UCC UCG UCU	ACA ACC ACG ACU	UGG	UAC UAU	GUA GUC GUG GUU	UAA UAG UGA
Ala	Arg	Asp	Asn	Cys	Glu	Gln	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val	stop
A	R	D	N	C	E	Q	G	H	I	L	K	M	F	P	S	T	W	Y	V	

Figura 2.4: Tabela de uso de códons padrão. A partir dela é possível ver a relação entre os aminoácidos e seus respectivos códons. Figura extraída de [AJL+08].

Como descrito em *Alberts et al.* [AJL+08], as proteínas são as moléculas mais estruturalmente complexas e sofisticadas funcionalmente conhecidas. Elas são os constituintes básicos dos organismos, fazem parte da maquinaria celular responsável pelo funcionamento celular e podem desempenhar diversas funções, como catálise de reações bioquímicas, estruturais e mecânicas. Duas ou mais proteínas podem ainda unir-se para desempenhar uma função, processo conhecido como interação proteína-proteína, e que faz parte de praticamente todos os processos celulares como, por exemplo, replicação do DNA, controle do ciclo celular, tradução de sinal, entre outros.

Os RNAs não traduzidos para proteína são definidos como RNAs funcionais (não codificadores). Tais RNAs, conhecidos como ncRNAs, estão envolvidos em diversas atividades celulares, como silenciamento de genes, replicação, regulação gênica, transcrição, e desempenham importantes funções, como regulação de expressão gênica e replicação [MLdPD08]. Na década de 1980 apenas se conheciam as famílias de *ncRNAs* de tRNA (RNAs transportadores) e de rRNA (RNAs ribossômicos, que são responsáveis pela síntese protéica). Hoje o número de famílias conhecidas ultrapassa 2000 (versão 11 do *RFAM*) [BDE+12]. Os *ncRNAs* são classificados de acordo com as seguintes categorias funcionais:

- *snRNA*: Estão envolvidos no processo de *splicing*;
- *snoRNAs*: Responsáveis por realizar modificações em outros *RNAs*;
- *RNasesP*: Responsáveis pelo processamento de *tRNAs*;

- *MRP*: Responsáveis pelo processamento de *rRNAs*;
- *miRNA*: Envolvidos com a regulação gênica pós-transcricional;
- *TERC*: Template de síntese telomérica;
- *tmRNA*: responsável pela liberação de *mRNA* defeituosos.

A figura 2.5 ilustra parte do dogma central de forma mais detalhada, no qual um gene produz uma proteína ou RNA funcional como produto final.

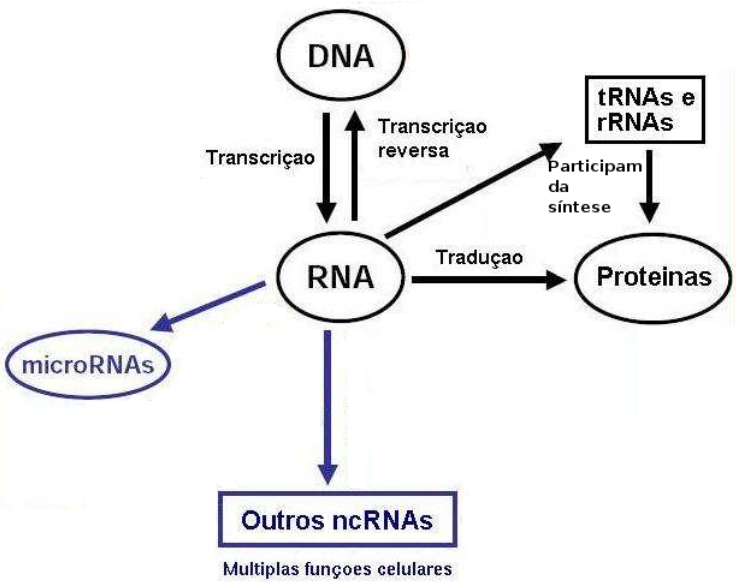


Figura 2.5: A figura ilustra o fluxo molecular da produção de proteínas ou *RNAs*. A partir de um gene (região do *DNA* é produzido uma molécula de *RNA* (transcrição), que por sua vez pode ser o produto final funcional ou produzir uma proteína (tradução). Figura modificada extraída de [ML06].

2.2 O problema da anotação

A evolução das tecnologias de sequenciamento de DNA tem permitido a elucidação da sequência genômica de um número cada vez maior de organismos. Contudo, a obtenção da sequência nucleotídica do genoma é apenas a primeira etapa no estudo dos organismos. É preciso identificar as diferentes regiões de interesse no genoma e suas funcionalidades, constituindo o que se denomina de processo de anotação [Lin01].

A primeira fase do processo de anotação consiste na identificação de elementos da sequência, regiões que podem ser identificadas a partir de determinadas características. Exemplos desses elementos são regiões repetitivas[1], genes de tRNA, mRNA e rRNA, genes codificadores de proteínas, entre outros. Existem vários programas para a identificação dos diferentes tipos de regiões genômicas. Para a predição de genes codificadores de proteínas, existem muitos programas disponíveis, como, por exemplo, o *GlimmerM* [DHK+99], *GlimmerHMM*[MPS04], *Glimmer3* [SDKW98], *Genscan* [BK97], *Phat* [CCS01], *Snap* [Kor04] e *Augustus* [SKG+06]. Para a localização de *RNAs* funcionais podemos citar o *RNAmmer*, que busca *RNAs* ribossômicos [LHR+07] e *tRNAscan-SE* [LE97], que busca *RNAs* transportadores. Para a identificação de repetições temos o *mreps* [KBK03] e o *TRF (Tandem Repeats Finder)*[Ben99], entre outros.

A segunda fase do processo de anotação consiste na coleta de evidências que poderão dar suporte a uma hipótese da função de cada uma das regiões identificadas. A forma mais simples é realizada a partir de busca de similaridade contra bases de dados de sequências conhecidas, utilizando-se mais comumente os programas do pacote *BLAST* [SGM+90]. Outra forma é a partir da execução de programas que buscam regiões conservadas, geralmente associadas com a função molecular. Por exemplo, para as regiões codificadoras de proteínas, podemos buscar motivos e domínios protéicos utilizando os programas *HMMER* [FCE11], *InterproScan* [HJM+11], *Phobius* [KKS04], *SignalP* [PBvHN11] e *TMHMM* [SvHK98].

A terceira fase do processo de anotação consiste na análise das evidências coletadas na fase anterior, a fim de se elaborar uma conclusão, que por sua vez representa uma hipótese sobre a possível função de cada elemento de sequência. Essa etapa pode ser realizada tanto de forma automática quanto manual. As ferramentas desenvolvidas para anotação manual exibem graficamente as regiões encontradas por processamentos anteriores. Ao se selecionar uma região, suas características e funcionalidades são exibidas, permitindo ao pesquisador analisá-la e gerar uma conclusão final. As ferramentas de anotação automática, por sua vez, conseguem atri-

[1]Regiões repetitivas são sequências que podem ter tamanhos variados e que podem ocorrer em várias localizações no genoma. As regiões repetitivas podem ser entremeadas ("interspersed") em múltiplos loci do genoma ou ser do tipo seriadamente repetidas ("tandemly repeated"). Nesse último caso, as repetições ocorrem em série, com um certo número de unidades repetitivas por locus, e um determinado período, que corresponde ao comprimento da unidade repetiva. Por exemplo, um locus contendo a sequência GACTGATCGATC tem três unidades repetitivas cujo período é de quatro bases (GATC). Aproximadamente 50% do genoma humano é composto de repetições [TS12].

buir possíveis funções às regiões , utilizando as evidências coletadas e inferindo as respectivas conclusões por meio de algoritmos pré-estabelecidos.

A maneira mais simples de se adicionar descrições às regiões é através de forma textual livre, de forma que as descrições são atribuídas de acordo com o especialista ou programa que realiza a análise. Essa forma de atribuição permite que termos distintos sejam utilizados para identificar regiões com as mesmas características, o que dificulta tanto a mineração das informações nas diversas bases de dados como pode causar ambiguidade na atribuição de novas conclusões. Nesse contexto, o uso de vocabulários restritos é muito mais adequado.

Um vocabulário controlado é a representação declarativa que inclui um conjunto fixo de termos para a representação das entidades de uma determinada área. Uma ontologia, além de conter a representação de um conjunto de termos, representa adicionalmente os relacionamentos entre esses termos [MEC07]. O processo de criação de vocabulários controlados para anotação ganhou força no final da década de 1990, quando se constatou a existência de um alto grau de conservação estrutural e funcional ao se comparar alguns genomas eucariotos, como *Saccharomyces cerevisiae*, *Caenorhabditis elegans* e *Drosophila melanogaster*. A partir dessa constatação, ficou claro que poderia-se aplicar, dentro de certos limites, a transferência de anotação entre espécies semelhantes, a chamada anotação transitiva [Con00]. Além disso, tendo-se dois ou mais organismos com genomas anotados, poderia-se realizar comparações entre suas anotações funcionais, evidenciando-se assim possíveis mecanismos moleculares para explicar diferenças fenotípicas. Para que essas abordagens pudessem funcionar adequadamente, foi necessário se criar ontologias, entre as quais se destacam o *Gene Ontology (GO)* [Con00] e *Sequence Ontology (SO)* [ELM$^+$05].

O consórcio *GO* é um projeto conjunto de três equipes de bases de dados de organismos modelo: *Flybase*, *Mouse Genome Informatics (MGI)* e *Saccharomyces Genome Database (SGD)*. Seu objetivo é a manutenção de uma ontologia para descrever as regras dos genes e seus produtos em qualquer organismo. O *GO* se estrutura em três categorias: processo biológico, função molecular e componente celular. *Processo biológico* diz respeito às atividades das células com as quais o produto do gene está relacionado. Estas envolvem transformação física ou química como, por exemplo, crescimento e manutenção celular (termo *GO: "cell growth and main-*

tenance"). *Função molecular* define a atividade bioquímica do produto do gene como, por exemplo, enzima (termo *GO: "enzyme"*) e transportador (termo *GO: "transporter"*). Por fim, *componente celular* refere-se à localização celular na qual o produto do gene está ativo como, exemplo, membrana nuclear (termo *GO: "nuclear membrane"*).

O *SO* é uma ontologia utilizada para descrever características e atributos de sequências biológicas. Seu objetivo é fornecer um conjunto padronizado de termos e relacionamentos para descrever anotações genômicas e prover uma estrutura necessária para o raciocínio automatizado sobre os seus conteúdos, de forma a facilitar a troca de dados e análises comparativas das anotações. O escopo do projeto *SO* é a descrição das características e propriedades das sequências biológicas. A figura 2.6 exemplifica um termo da ontologia e seus relacionamentos com os demais termos.

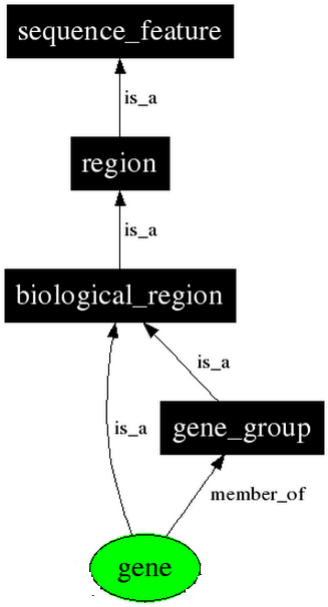


Figura 2.6: Exemplo de um termo de ontologia da sequência e seus relacionamentos. Uma *gene* é membro de um grupo de genes (*gene_group*) e uma região biológica (*biological_region*), que é uma região (*region*) que, por sua vez é *feature_sequence*.

Tanto o *GO* quanto o *SO* são ontologias abertas e em contínua evolução, permitindo assim que novos termos e definições, bem como relacionamentos, possam ser propostos. Esses novos termos são debatidos e podem ser aprovados ou rejeitados pelo grupo que administra as ontologias. Finalmente, para registro do resultado final de anotações existem vários formatos padronizados. Os mais utilizados são: *Feature Table* [fea12], *GenBank* [BKMC$^+$11], *GFF3*

[Ste13], *GTF* [gtf13] e *EMBL* [emb05].

2.3 Esquemas de bancos de dados

A evolução das tecnologias de sequenciamento aumentou consideravelmente o número de genomas sequenciados. O advento dos sequenciadores de segunda (ou nova) geração diminuiu os custos e ampliou dramaticamente a velocidade do sequenciamento, permitindo o estudo de cada vez mais sequências. Essa mudança na escala da anotação fez surgir a necessidade da utilização de banco de dados para armazenamento dessas informações.

A fim de facilitar a representação dos dados biológicos nos bancos de dados, foram desenvolvidos esquemas para representar de forma mais fiel as informações biológicas. Aqui descreveremos três deles: *GUS* [CJKP05], *Chado* [MEC07] e *Biosql* [Lap01]. Os dois primeiros são amplamente utilizados na literatura[2] e o último foi desenvolvido com o objetivo de ser uma camada de armazenamento de dados comum, apoiada pelo projeto *Bio**, que consiste de um conjunto de voluntários que colaboram para prover funcionalidades nas áreas de bioinformática, genômica e ciências da vida a partir de licenças de softwares de código aberto (*open-source software - OSS*) [PGY+12].

2.3.1 Biosql

O biosql [Lap01] é um esquema para armazenamento de sequências, suas características, anotações, referência taxonômica e ontologias, que faz parte do projeto *GMOD (Generic Model Organism Database project)*[3] [gmo08]. O esquema foi criado em 2001 por Ewan Birney, como armazenamento relacional local para o repositório de sequências *GenBank* [BKMC+11], e se tornou uma colaboração entre os projetos (*BioPerl* [4], *BioJava* [5], *BioPython*[6] e *Bioruby*[7]). Seu objetivo é construir um esquema suficientemente genérico para o armazenamento persistente

[2]Por meio do web site *Web of Knowledge*[web13] buscamos o número de vezes cada um dos esquemas foi citado na literatura. O *Chado* possui aproximadamente 60 citações e o *GUS* 300.

[3]O GMOD é uma coleção de ferramentas de software de código aberto para criar e gerenciar bancos de dados biológicos em escala genoma.

[4]http://www.bioperl.org/

[5]http://biojava.org/

[6]http://biopython.org/

[7]http://bioruby.org/

de sequências, recursos e anotação de uma forma interoperável entre os projetos *Bio**. Cada projeto *Bio** possui um mapeamento para *BioSQL*. Isso significa que os dados são facilmente manipulados entre os diferentes projetos. Por exemplo, dados armazenados no banco por um programa escrito em *BioPerl* podem ser facilmente recuperados por um programa escrito em *BioJava*.

O biosql é um esquema conciso, desenvolvido com o objetivo de ser suficientemente genérico para o armazenamento persistente das sequências, suas características e anotações de diferentes fontes, como os repositórios *GenBank* ou *SwissProt*. Possui apenas três módulos: informações gerais, informações da sequência e termos de ontologia, que somam 27 tabelas. O esquema faz distinção entre a sequência e suas características, mas suas características são tratadas como *SeqFeatures*, e tipificadas por termos de ontologias. É possível localizar uma característica na sequência a partir de um sistema de coordenadas e estabelecer relacionamentos entre características.

2.3.2 GUS

O *GUS (Genomic Unified Schema)* [CJKP05] é um esquema de banco de dados biológicos, desenvolvido pela Universidade da Pensilvânia e capaz de representar uma grande quantidade de tipos de dados, incluindo dados genômicos, expressão gênica, montagem de transcritos, proteômica, entre outros. Em conjunto com o *framework* de aplicação *GUS* e o kit de desenvolvimento Web *GUS (GUS Web Development Kit (WDK))*, constitui um sistema modular de armazenamento, integração de dados e análise funcional de dados genômicos [Iba03]. Atualmente, é utilizado por diversos projetos como *GeneDB, CryptoDB, ApiDB, PlasmoDB, BiowebDB*, entre outros. O GUS é formado por cinco esquemas: banco de dados de sequências transcritas (*DoTS - Database of Transcribed Sequences*), banco de dados de abundância de *RNAs* (expressão de gene e experimentos de microarranjo, *RAD - RNA Abundance Database*), sistema de busca de elementos de transcrição (*TESS - Transcription Element Search System*), recursos compartilhados (*SRes - Shared Resources*) e esquema central. Possui uma estrutura detalhada, com tabelas distintas para o registro das análises efetuadas pelos vários *softwares* utilizados no processo de anotação. Isso torna a sua estrutura bastante complexa, com mais de 400 tabelas, o que difi-

culta sua compreensão e torna complexa a tarefa de inclusão de novas análises em processos de anotação automática.

2.3.3 Chado

O esquema Chado [MEC07] parte de uma estratégia de modelagem diferente do *GUS*. A idéia é criar uma forma de representação abstrata das características e análises das sequências, a partir da natureza da informação. Baseado nisso, esse esquema é capaz de representar um conjunto mais conciso de tabelas, resultado das mais distintas análises oriundas dos processos de anotação, assim como fenótipos, genótipos, ontologias, publicações e filogenia. O Chado é utilizado por muitos projetos como *FlyBase*, *Xenbase*, *ParameciumDB*, *IGS*, *AphidBase, BeeBase, BeetleBase, BovineBase*, entre outros. O esquema é composto por 8 módulos principais: *Companalysis* (dados de análises computacionais), *Controlled Vocabulary* (vocabulários controlados e ontologias), *General* (identificadores), *Genetic* (dados genéticos e genótipos), *Map* (mapeamentos sem sequência), *Organism* (dados taxonômicos), *Publication* (publicações e referências) e *Sequence* (sequências e características das sequências). Ao todo, o esquema possui um pouco mais de 170 tabelas.

O *Chado* é um esquema dirigido por ontologias. Em outras palavras, faz uso de ontologias não somente para anotar entidades biológicas como os outros esquemas, mas também para tipificação de entidades e de relacionamentos entre as elas. Essa característica dá ao esquema uma grande flexibilidade, uma vez que não é preciso ter tabelas específicas para os diferentes tipos de entidades.

O esquema *Chado* é baseado na sequência. O módulo *Sequence*, mais especificamente a tabela *feature*, é central para o gerenciamento de dados das sequências. O esquema define uma *feature* como uma região de uma macromolécula biológica (*DNA, RNA* ou polipeptídeo) ou um agregado de regiões desse polímero. As *features* são tipadas de acordo com um termo de ontologia, armazenado no módulo de vocabulário controlado. As associações entre as *features* são estabelecidas através da criação de relacionamentos entre elas. Por exemplo, uma região é parte de uma sequência maior. Os relacionamentos são igualmente tipificados por termos de ontologia. Adicionalmente, as *features* podem ser localizadas em relação a uma outra *feature*,

a partir de um sistema de coordenadas.

2.3.4 Discussão

Os esquemas descritos são capazes de armazenar informações a respeito das sequências e suas características, a diferença entre eles está na forma como tais informações são armazenadas. Enquanto o *GUS* possui um extenso conjunto de tabelas para representar os diferentes tipos de características das sequências, o *Chado* e o *Biosql* possuem uma representação genérica. O primeiro trata a sequência e suas características de forma abstrata, pois considera que ambas são a mesma entidade, tipadas a partir de termos de ontologia, e o segundo considera que apenas as características são abstratas e tipadas por ontologias. A associação entre as sequências, no *Chado* e entre as características, no *BioSql*, é estabelecida por relacionamentos e por localização de uma sequência em relação à sua referência. Os três esquemas são capazes de armazenar as anotações atríbuídas as sequências.

O *GUS* e o *Chado* possuem estruturas para registrar tanto as análises realizadas bem como os resultados encontrados, a diferença entre os dois esquemas é que o *GUS* possui tabelas específicas para cada tipo de análise, enquanto o *Chado* trata análise como uma entidade genérica, caracterizada pelos seus atributos e propriedades. O *Biosql* não possui a capacidade de armazenamento das análises, sendo capaz de armazenar apenas as sequências e suas características.

Para representação dos dados em nossa plataforma, escolhemos o esquema *Chado* devido à sua capacidade de representar de forma genérica as análises realizadas realizadas, ser baseado na sequência, dirigido por ontologias e modular.

2.4 Ferramentas de anotação

Vários programas foram desenvolvidos com o objetivo de realizar tarefas relativas ao processo de anotação. Tais programas podem executar uma tarefa específica relativa a uma das três fases do processo de anotação ou integrar programas para um conjunto de tarefas. De acordo com suas funcionalidades, essas ferramentas foram classificadas como *pipelines*, construtores de fluxo de trabalho, visualizadores, anotadores e ferramentas integradas. Nesta seção tratare-

mos individualmente cada tipo.

2.4.1 *Pipelines*

Pipelines são processos pelos quais dois ou mais programas podem ser executados de forma coordenada, em uma determinada ordem, onde a saída de cada um é redirecionado como entrada do próximo. Em geral, os *pipelines* de anotação genômica são conjuntos de programas capazes de identificar as regiões genômicas, atribuir funcionalidades as mesmas e devolver os resultados para o usuário em um formato de arquivo de anotação. A fim de ilustrar a idéia do processo, descreveremos nessa seção quatro *pipelines* de anotação, onde três são utilizados para anotação de genomas e um para anotação de proteínas.

O *Genescript* [HCBS02] é um *pipeline* de anotação de sequências de *DNA* desenvolvido pelo *Centre for Applied Genomics*, composto por programas para a identificação de regiões genômicas (*RepeatMasker* [SHGnd] para predição e mascaramento de regiões repetitivas e *GenScan* [BK97], *HMMgene* [Kro97] e *GRAIL-EXP* [HSS$^+$00] para predição de genes) e atribuição de funcionalidade por busca de similaridades (*Blast* [SGM$^+$90]) contra diversas bases de dados externas. O resultado do processamento é devolvido nos formatos *GFF, Genbank, EMBL*, VISTA e FASTA.

O *pipeline DIYA (Do-It-Yourself Annotator)* [SOR09] foi desenvolvido para anotação de genomas procarióticos, com o objetivo de identificação das funções dos genes e realização de análises comparativas, que aceita como entrada genomas inteiros ou trechos sequenciados (*contigs*). Para a identificação das características das sequências utiliza os programas *Glimmer3* [SDKW98] (predição de genes) e *tRNAScan* [LE97] (identificação de *tRNAs*) e para atribuição de funcionalidade por busca de similaaridade com os programas *BLAST* e *RPS-BLAST* como padrão da ferramenta, contudo usuário pode modificar os programas que realizam essas análises. Os resultados dos processamento são retornados nos formatos *GenBank* ou *GFF*.

O *JCVI (J. Craig Venter Institute)* [TGM$^+$10] é um *pipeline* para a anotação de dados de sequenciamento metagenômico, que foi desenvolvido com o objetivo de receber como entrada tanto um conjunto de dados na magnitude de milhões de sequências como sequências individuais de organismos procariotos. O *pipeline* é estruturalmente dividido em componentes de

anotação estrutural e funcional, que podem trabalhar sozinhos ou em conjunto. Os componentes estruturais tem por objetivo identificar regiões codificadoras de proteínas (programa *MetaGeneAnnotator* [NTI08]) e *RNAs* funcionais (*tRNAScan-SE* e *BLAST* contra banco de dados de sequências de *RNAs* funcionais). Os componentes funcionais executam um conjunto de programas, como o *BLAST* sobre as regiões codificadoras encontradas na etapa anterior com o objetivo de identificar suas funções.

O *PIPA (Pipeline for Protein Annotation)* [YZD+08] é um *pipeline* para a anotação de função de proteínas, que consiste na combinação de diversos programas e bases de dados externos. O *pipeline* recebe como entrada sequências protéicas em formato *FASTA*, realiza o processamento para identificação de funcionalidade, converte os resultados encontrados para termos da ontologia de gene (*GO*), gera um consenso a partir dos resultados encontrados e os devolve para o usuário em formato de arquivo de anotação *GFF*.

Discussão

Como observado, os *pipelines* descritos nessa seção compartilham de um conjunto de características: possuem um fluxo de execução padrão e um conjunto fixo de programas (a exceção é o *pipeline DIYA*, pois permite que o usuário modifique seus programas) e trabalham apenas com arquivos (não estão conectados a bancos de dados).

2.4.2 Construtores de fluxo de trabalho

Os construtores de fluxo de trabalho são sistemas que permitem ao usuário montar um esquema que executa uma série de tarefas para o processamento de suas sequências, que podem gerar dois tipos de fluxos, os fluxos de trabalhos genéricos (*workflows*) ou os *pipelines*. A primeira categoria permite a criação de um conjunto amplo de tarefas, uma vez que não exige a conexão física entre seus componentes, apenas a ordem de execução dos componentes é definida. Uma desvantagem é que esse processo pode requerer um maior conhecimento computacional, pois é possivél que haja a necessidade de conversão entre as entradas e saídas dos programas. Nos *pipelines*, por terem etapas fixas, tais conversões já estão implementadas. A partir da revisão da literatura foi possível encontrar construtores de fluxo como o *EGene* [DKM+05], *Pegasys*

[SHS+04], o *Bioinformatics Computational Journal (BCJ)* [FRG+07], o *BioWMS* [BCMS07], o *Cyrille2* [FvdBD+08], o *Taverna* [OAF+04] e o *Biopipe* [HRmC+03].

O *eGene* [DKM+05] é um sistema integrado para a construção de *pipelines* com o objetivo de processar e anotar automaticamente sequências biológicas, que consiste de um repositório de componentes de processamento e um arcabouço para a execução de *pipelines*. Permite que o usuário construa seu *pipeline* a partir de um conjunto de componentes de acordo com sua necessidade. O *EGene* é composto por mais de cinquenta componentes, que englobam as tarefas de pré-processamento, coleta de evidências para anotação e construção de relatórios de anotação. Cada componente é um *script Perl*, que pode funcionar como um componente autônomo ou utilizar *softwares* de anotação disponíveis. Quando trabalha em conjunto com os *softwares*, o *script* é implementado de forma a executar os programas, parsear os resultados e integrá-los ao sistema. O elemento-chave que garante a abstração da representação é o componente SequenceObject.pm, um módulo *Perl* que encapsula todas as funcionalidades relacionadas com a manipulação de sequências. Um *pipeline* é descrito através de um arquivo de configuração, que consiste de um arquivo texto que especifica, em ordem, os programas a serem executados e seus respectivos parâmetros. Tal arquivo pode tanto ser criado a partir de editores de texto como através da ferramenta gráfica do *EGene*, o *CoEd*. Desenvolvido em *Java*, o *CoEd* especifica todos os parâmetros de entrada de um programa (obrigatórios e opcionais), e atribui os valores-padrão aos mesmos (que podem ser modificados pelo usuário). A figura 2.7 ilustra um *pipeline* construído a partir do *Coed*. Os retângulos representam os componentes escolhidos e as setas a ordem de execução. Para a execução de um *pipeline*, um programa em *Perl* lê o arquivo de configuração e inicia cada etapa de processamento. Os resultados são devolvidos nos formatos de arquivos padrão, como *Feature Table*, *GenBank* e *GFF*. Um outro formato de saída gerado é o *XML*, um formato particular do sistema, que pode ser utilizado como entrada para a execução de um novo *pipeline* do *EGene*.

O outro tipo de sistemas construtotes de fluxo de trabalhos geram *workflows*, que são sistemas poderosos uma vez que descrevem um amplo conjunto de tarefas a partir da inclusão de vários *softwares*. Contudo os *workflows* são mais difíceis de entender e de se expandir, pois na maioria dos sistemas existe pouca abstração em cada módulo. Em geral os *workflows* possuem

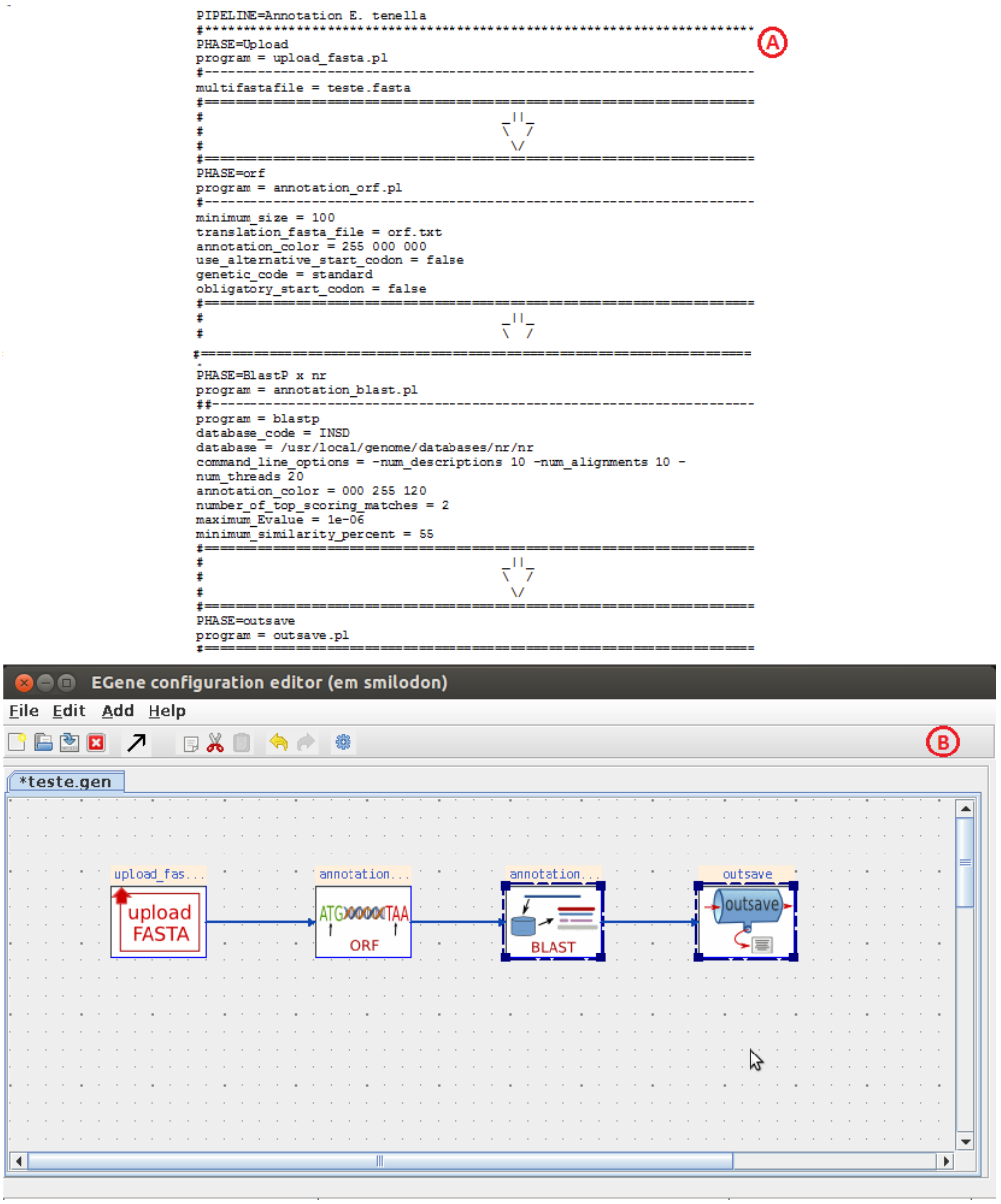


Figura 2.7: Exemplos de um *pipelines* do *EGene*. A parte de cima da figura (A) ilustra um *pipeline* construído através de um editor de texto e a de abaixo (B) mostra o mesmo *pipeline* representado no *CoEd*. Os retângulos representam os componentes a serem executados e as setas estabelecem a ordem de execução.

um componente capaz de realizar conversões entre os arquivos utilizados por alguns programas, o que restringe o usuário a adicionar ao seu *workflow* somente programas que utilizem arquivos nos formatos compatíveis às conversões realizadas. Como o foco do nosso trabalho é a busca de um sistema que permita ao usuário montar seus fluxos de trabalho livremente, não nos focaremos no estudo dos *workflows*.

2.4.3 Visualizadores

Os visualizadores de anotação são ferramentas que permitem ao usuário analisar graficamente as característícas de um genoma. Tais características são representadas como figuras geométricas em um sistema de coordenadas, onde o eixo horizontal representa a sequência e cada tipo de característica é exibida em diferentes níveis e cores, de forma a diferenciá-la das demais. As características que atríbuem valores a cada posição da sequência, como conteúdo *GC*, são representadas como gráficos. O visualizador permite o usuário navegar na sequência, buscar uma região específica (a partir de suas coordenadas), variar o nível de detalhamento, bem como visualizar as anotações atribuídas às regiões.

Descreveremos nessa seção três visualizadores disponíveis na literatura: *GBrowser* [SMS+02], *JBrowser* [Sea09] e o *Gaggle* [BKR+10].

GBrowser

O *GBrowser (Generic Genome Browser)* [SMS+02] é um visualizador de genomas desenvolvido pelo *GMOD* [gmo08], que consiste na combinação de uma base de dados e páginas web interativas para a visualização das regiões genômicas de uma sequência e das anotações atribuídas às regiões. O *GBowser* aceita como entrada tanto arquivos em formatos de anotação quanto conexão com um banco de dados por intermédio de um conector (embora os próprios autores citem queda no desempenho) e é capaz de exibir sequências de diversos tamanhos - desde pequenas até as que possuem megabases de tamanho.

A funcionalidade de busca do navegador é bastante flexível. A pesquisa na sequência pode usar um conjunto diversificado de parâmetros como nome do cromossomo de acordo com o organismo que está sendo exibido (por exemplo, *II* para *C. elegans*, *2L* para *D. melanogaster* e 2 para *Mus musculus*), nome do *contig*, identificador do repositório GenBank, entre outros.

O *GBrowse* permite ao usuário adicionar anotações particulares a respeito da sequência exibida. Para tanto, o usuário deve criar um arquivo em formato *GFF* que contém suas anotações e a localização das mesmas na sequência. Adicionalmente o usuário configura os atributos gráficos das anotações, como cor, altura, entre outros. A figura 2.8 ilustra a tela principal do visualizador.

A ferramenta permite que o usuário configure o navegador, a fim de definir quais tipos de características da sequência devem ficar visíveis, a ordem de exibição na tela e as cores das características. As configurações de cada usuário ficam armazenadas, de forma que são recuperadas toda vez que o usuário acessa o navegador.

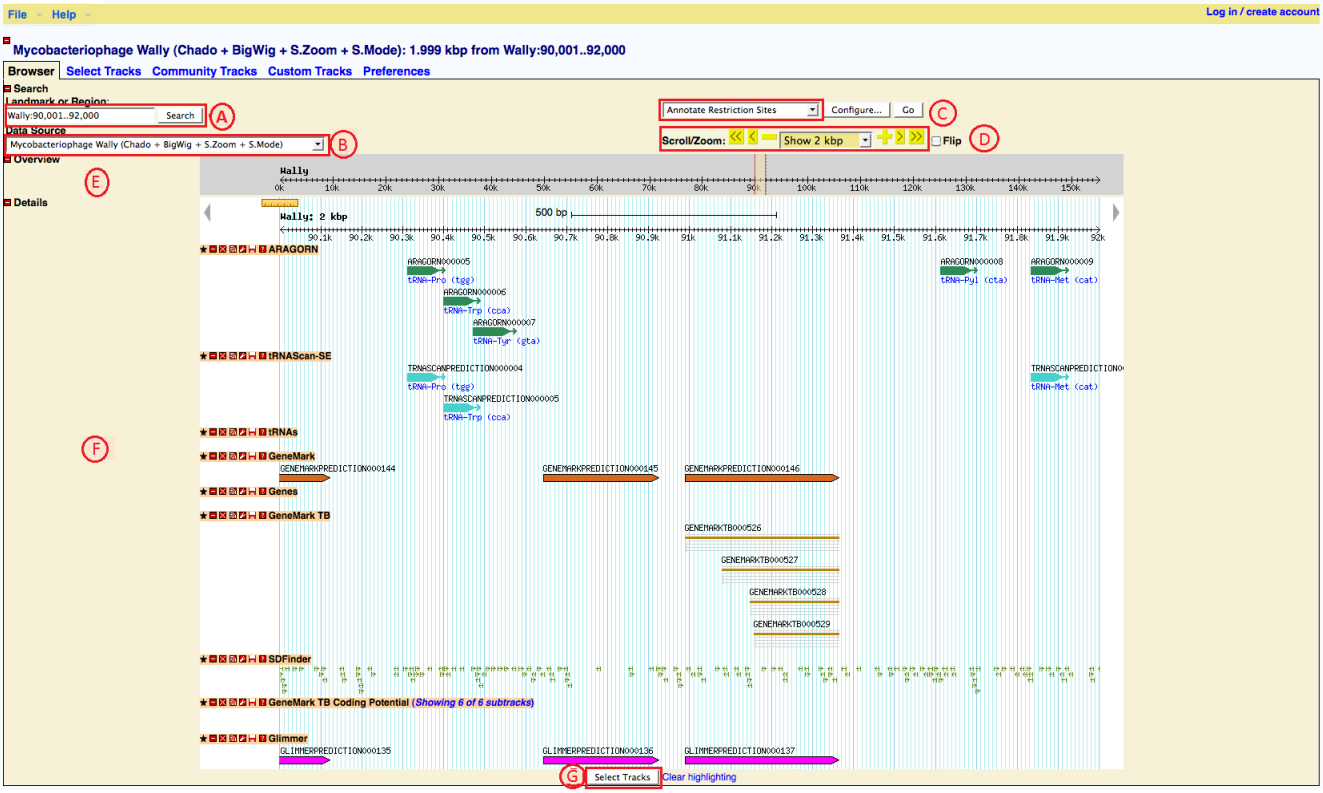


Figura 2.8: Tela principal do navegador *GBrowse*. No canto superior temos duas caixas, uma para limitação da região de busca (A) e outra para seleção das sequências (B). O navegador permite ao usuário configurar as características exibidas no navegador (C). A ferramenta disponibiliza a capacidade de navegar no genoma e de alterar o nível de detalhamento da sequência (D). A sequência é representada como um eixo horizontal (E) e suas características são exibidas abaixo do eixo e agrupadas por tipo(F). O usuário pode configurar quais caracteríticas devem ficar visíveis no navegador a partir da opção *Select Tracks* (G).

Caso o usuário deseje analisar as propriedades de uma característica, basta selecioná-la e o navegador exibirá uma tela com as devidas descrições. A figura 2.9 ilustra o exemplo de uma característica selecionada.

JBrowser

O *JBrowser* [Sea09] é um visualizador de genomas baseado em *JavaScript*. Da mesma forma que o navegador anterior, o *JBrowser* aceita como entrada tanto arquivos em formato de

Name:	**s02**
Class:	Sequencia
Type:	Sequencia
Source:	exemplo
Position:	ctgA:24562..28338 (+ strand)
Length:	3777

>s02 class=Sequencia position=ctgA:24562..28338 (+ strand)
ctgcctaccgggtcgaattatttacgcgtgttacaatatgtaatttagaaaaagggattgctggtcgatgcgtctccaag
ggatttttatctaaaagcatccttttgggtgtactctgatcgcacgtcgcagacagcagtgggttttgacgcagtccgt
aggcccacagactcgtttgttgtttattaatcccaggggagcgttgaagccacacctattctgtagctgtttgaaaggta
gctagcccggatattactcaaggtgactcccttcagaatcacacgtcgctggagtcgccacagggtggcatatacgagtg
atagagcaccttactttcgaggtagcggtacattagtgcaacgatgaacccactatagtcttagtgatttcatgttttac
ttacgcgaaaacgtgggggttttgtcaacacgtatacgttgaatgcacatgcctcatcctaaactgatgcactgccacaag
tctgaaagagcgacagtctgcaacatagcggaaggttacgcccaagccagtggtgatcccccataagcttggagggactc
cccttagcgttggatgtctttgccccagcggcctcggtgtacgggttctccaccccactatggtttggaactatgaagag
gtacggcaacctacccgaggcaccaaatcgtgaacctacgcctatatatacggatagcagggtatccattcttaccatga
gctcgtaaaccactccgctgaatttcgatgggctttggcgcacatcaccgtttctatcacagatctgtcaacggaatctaa
cgctatttactcggcgcacacagatcggaaaacccaactgtggcgcgggacggactccaggaatcgttacgcgttatcac

Figura 2.9: Tela que exibe as propriedades de uma característica selecionada no *GBrowser*. Nela são exibidas informações como nome, tipo, fonte, localização, tamanho e composição nucleotídica.

anotação como conexão com um banco de dados.

A funcionalidade de busca dessa ferramenta é um pouco mais restrita que a do *JBrowser*, pois permite a busca apenas pelo nome de uma característica ou seu identificador, cromossomo e /ou coordenadas da sequência.

O *JBrowser* permite ao usuário configurar quais características devem aparecer e sua respectiva ordem no visualizador. Essa configuração é feita facilmente a partir da ação de selecionar e arrastar a característica na tela. A figura 2.10 mostra a interface visual da ferramenta.

De forma semelhante ao *GBrowse*, o navegador permite ao usuário visualizar as informações de uma característica seleciona. Para tanto o usuário deve selecionar a característica de interesse com o botão esquerdo e escolher a opção *View details*. Nessa ferramenta, as propriedades são exibidas em uma tela distinta. A figura 2.11 mostra um exemplo das informações de uma característica selecionada.

Nossa pesquisa bibliográfica indicou que o *JBrowse* é ainda pouco utilizada em projetos (apresenta-se pouco citada em trabalhos). O *GBrowser* ainda é preferida como ferramenta de visualização em projetos genômicos.

Gaggle

O *Gaggle Genome Browser (GGB)* é uma ferramenta gráfica interativa desenvolvida em Java (diferente do *GBrowser* e do *JBrowser*, esse visualizador não exibe as sequências em uma página web). O visualizador foi desenvolvido com o objetivo de permitir exploração interativa

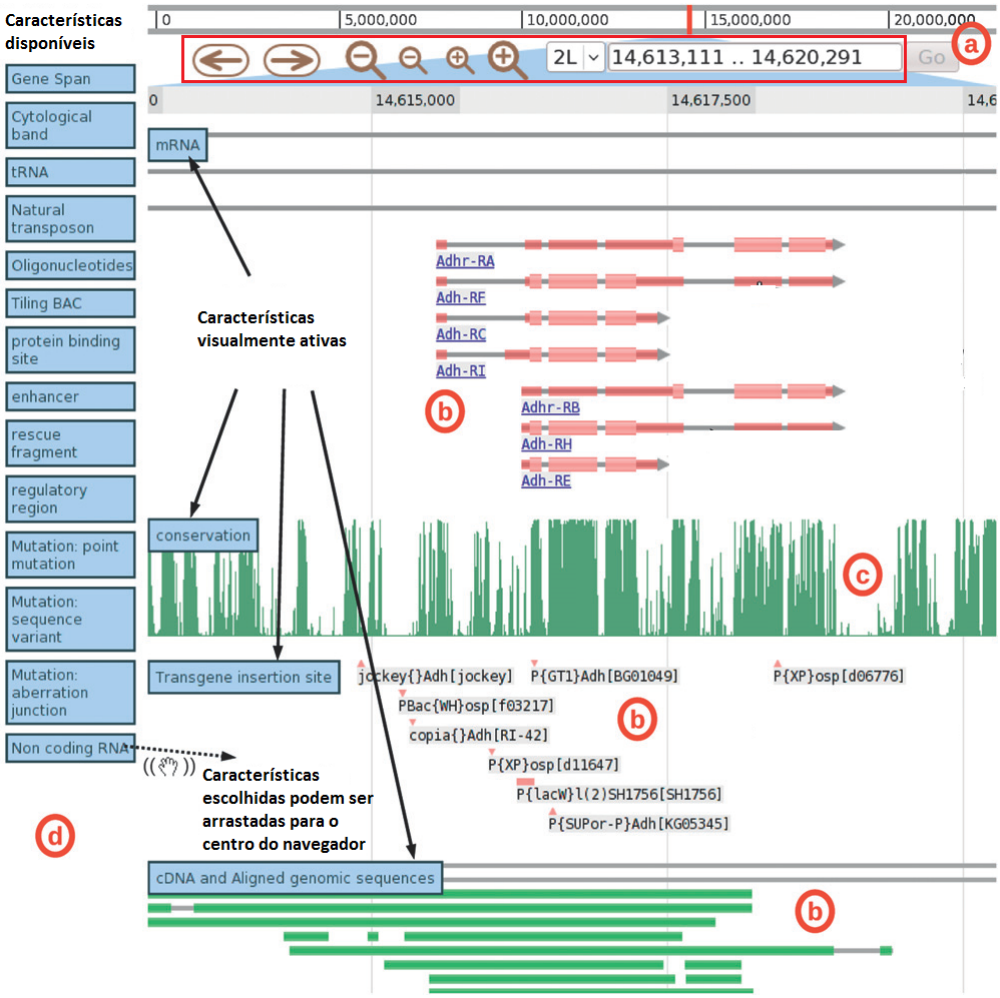


Figura 2.10: Visualização das características de uma sequência a partir do *JBrowser*. A ferramenta permite ao usuário navegar na sequência, seleciona níveis de zoom, selecionar um cromossomo e uma região a partir das suas coordenadas (A). As características da sequência são exibidas a partir de figuras geométricas no centro da tela(B). As características quantitativas da sequência são exibidas como gráficos (C). No exemplo são exibidas regiões de conservação. As características da sequência que não estão sendo exibidas no momento ficam são listadas à esquerda da tela (D). Para exibir alguma delas, basta selecionar a desejada e arrastar para o centro do navegador. Figura modificada extraída de [Sea09].

e ter fácil acesso aos dados, com a habilidade de manipular grandes conjuntos de dados. A pesquisa por características na ferramenta é feita por meio de palavras-chave.

Diferente dos visualizadores descritos, o *Gaggle* não se conecta com algum banco de dados. A exibição das sequências é feita através da entrada de arquivos. O usuário pode configurar quais características devem ser exibidas e sua forma de exibição (como cor, altura) e importar novas características particulares em formato *GFF*. A figura 2.12 ilustra a interface gráfica do *Gaggle Genome Browser*.

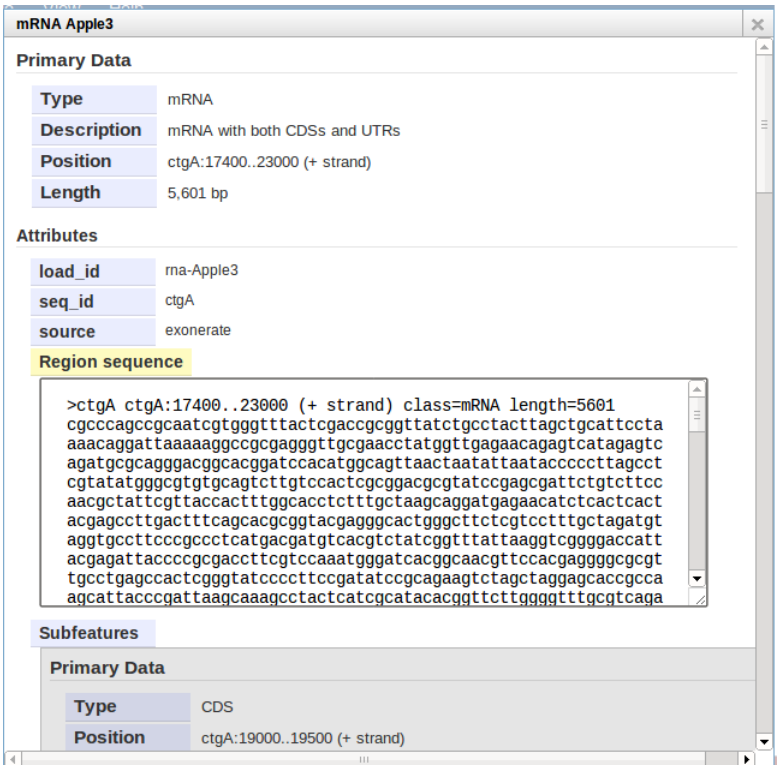


Figura 2.11: Tela secundária do *JBrowser*. Ao selecionar uma característica da tela principal, o navegador abre uma tela com as informações a respeito da característica selecionada.

Apesar de ser um visualizador, permite que informações em forma de texto sejam adicionadas às regiões, que são anexadas na forma de *bookmarks*. A figura 2.13 ilustra a tela de anotação do *Gaggle*. Ao selecionar a região, são retornadas sua localização genômica e sequência. A caixa de texto *Annotation* permite que o usuário digite as informações que deseje. A ferramenta disponibiliza a opção de executar uma busca de similaridade a partir da sequência de bases da região selecionada. Ao selecionar essa opção, a ferramenta abre um navegador com a página da ferramenta *BLAST* online juntamente com a sequência nucleotídica a ser buscada. Adicionalmente, é possível exportar as informações atribuídas a sequência em formato de arquivo de texto.

Discussão

Os visualizadores descritos nessa seção trazem algumas semelhanças e diferenças entre si. Em relação à entrada de dados, o *GBrowser* e o *JBrowser* aceitam tanto dados em formatos de aquivos de anotação como conexão com bancos de dados. Já o *Gaggle* apenas trabalha

Figura 2.12: Interface visual do *GGB*. Diferente dos visualizadores descritos anteriormente, o eixo que representa a sequência se encontra no centro do visualizador (A). As características localizadas acima do eixo se relacionam com a fita direta e as abaixo com a fita complementar. A barra lateral à esquerda (B) ilustra os cursores disponíveis para seleção e a barra de zoom. Através da ferramenta de busca, o usuário pode pesquisar uma característica a partir de uma palavra-chave(C). O usuário pode adicionar informações às regiões das sequências, que são listadas à direita do visualizador(D). Ao selecionar característica, informações a seu respeito são exibidas em um painel (E).

com manipulação de arquivos. Quanto à entrada de arquivos anotações particulares, apenas o *GBrowser* e o *Gaggle* permitem que o usuário entrem com essas informações. Em relação à seleção das características que devem ficar visíveis no navegador, todos os visualizadores permitem o usuário configurar quais deve aparecer. Destaque para o *JBrowser*, uma vez que esta ação pode ser feita facilmente (basta selecionar a característica escolhida e arrastá-la para o centro da tela). Com relação à visualização das características, a exibição no *Gaggle* é feita de forma diferenciada do *GBrowser* e do *JBrowser*, uma vez que o eixo é representado no centro da tela e as características relativas a fita direta são exibidas acima do eixo e as da fita complementar abaixo da tela.

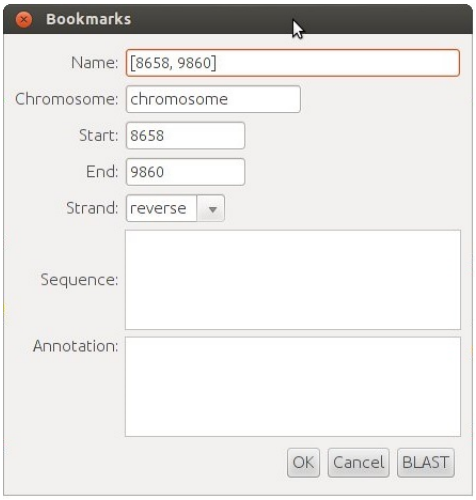


Figura 2.13: Tela de atribuição de informação à uma característica do *Gaggle*. Ao selecionar uma região, são exibidas sua localização genômica (cromossomo, fita, início e fim) e o conjunto de bases da região. Caso o usuário se interesse em realizar um *BLAST*, a ferramenta abre um *web browser* com a ferramenta online.

2.4.4 Editores de anotação

Editores de anotação são ferramentas que recebem como entrada sequências previamente processadas, exibem graficamente as evidências encontradas e anotações atribuídas e permitem a alteração de informações da sequência. De forma similar aos visualizadores, as características da sequência são apresentadas como figuras geométricas que se localizam em um sistema de coordenadas, onde o eixo representa a sequência referência. Da mesma forma que no *Gaggle*, o eixo de coordenada nos anotadores se localiza no centro da tela e as figuras localizadas acima do eixo se relacionam com as características da fita direta e as abaixo com a fita complementar. As características são diferenciadas umas das outras pela cor, muitas vezes configurável pelo usuário. As informações a serem atribuídas às regiões dependem das permissões de cada anotador. Como anotadores podemos citar o *Artemis* [RPC$^+$00], o *Apollo* [LSH$^+$02] e o *GenDB* [MGM$^+$03]. O *GenDB*, apesar de não ser somente um editor, exibe as graficamente os resultados dos processamentos e permite que o usuário adicione informações manualmente às sequências.

Artemis

O *Artemis* [RPC+00] é uma ferramenta de visualização e anotação manual de sequências tamanho pequeno e médio desenvolvida em *Java* pelo Instituto *Sanger* [8] que permite ao usuário navegar interativamente pela sequência e visualizá-la em diferentes níveis de granularidade (desde aminoácidos até o genoma completo). A ferramenta possui navegação interativa, capacidade de representar graficamente certas propriedades das sequências (como conteúdo *C+G*, *skew C/G*, códons de uso, entre outras), visão estatística da sequência (por exemplo, porcentagem de determinada base na sequência ou em certa região dela) e visualização de propriedades individuais de proteínas, como hidrofobicidade, que pode trabalhar tanto com a manipulação de arquivos como conectada a um banco de dados com uma modelagem *Chado*. Em relação aos arquivos, aceita como entrada sequências nos formatos EMBL, Genbank, GFF3 e FASTA. Em relação ao número de sequências que podem ser abertas simultaneamente no *Artemis*, quando essa ferramenta trabalha com a manipulação de arquivos, mais de uma sequência pode ser visualizada. Contudo, quando o *Artemis* está conectada a um banco, apenas uma sequência pode ser analisada por vez.

Em relação às sequência analisada, a ferramenta permite ao usuário executar um conjunto de ações, como adicionar, editar e apagar características. Adicionalmente, o *Artemis* permite ao usuário selecionar uma característica da sequência e executar buscas de similaridade *online* contra base de dados de sequências conhecidas. Para tanto, quando o usuário seleciona essa opção, o *Artemis* abre um navegador com site do *Blast* (http://www.ncbi.nlm.nih.gov/blast) passando como entrada a sequência nucleotídica da região selecionada juntamente com os parâmentros ajustados pelo usuário e executa a busca. Os resultados são exibidos na própria página do *Blast*. O *Artemis* permite a adição de programas externos, contudo essa tarefa não é simples, uma vez que exige um maior conhecimento na estrutura de ambas as ferramentas para que se possa implementar *scripts* para a troca de informações entre mesmas. A figura 2.14 ilustra a tela principal do *Artemis*.

A adição de informações às regiões das sequências é feita de duas formas, dependendo da maneira que o usuário está trabalhando (manipulação de arquivos ou conectado a um banco de

[8] http://www.sanger.ac.uk

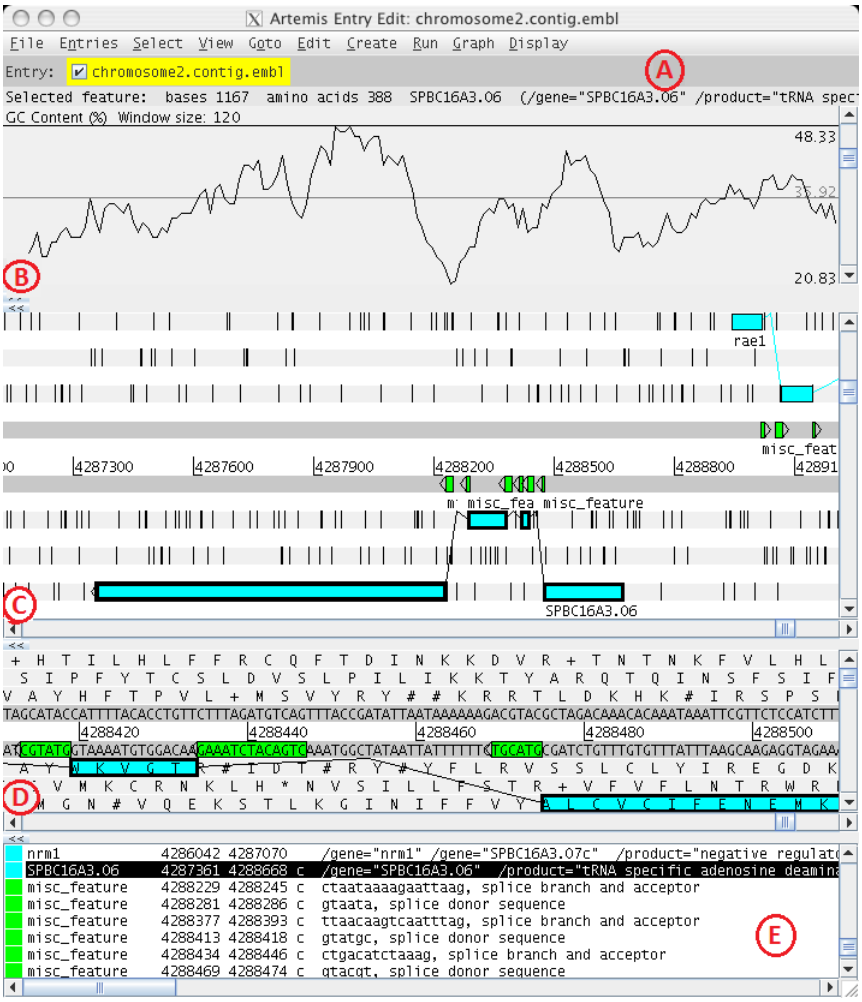


Figura 2.14: Tela principal do anotador. O *Artemis*, quando trabalha com a manipulação de arquivos, permite ao usuário abrir mais de uma sequência de uma só vez. As sequências carregadas no sistema são exibidas para o usuário (A) e as atualmente visíveis estão selecionadas. As informações quantitativas são exibidas através de gráficos nas primeiros níveis da tela (B). As características das sequências são exibidas em dois níveis de detalhamento: sem (C) e com os seus nucleotídeos visíveis (D). No final do anotador são listadas todas as características da sequência, com suas coordenadas e informações anexadas (E).

dados). Quando trabalha com manipulação de arquivos, a anotação do *Artemis* é feita por meio de texto livre, na qual o usuário seleciona um qualificador (*qualifier*) e adiciona os termos de acordo com a sua preferência. A figura 2.15 ilustra a tela de anotação do *Artemis*. Quando o *Artemis* está conectado a um banco de dados, o ambiente de anotação exibe a região selecionada de forma detalhada. A adição de termos pode ser realizada tanto de forma textual, como a partir da seleção de termos de ontologia disponíveis na base de dados e por meio de adição de ortologia e parologia com outras sequências. A figura 2.16 ilustra o ambiente de anotação da ferramenta quando conectada a um banco de dados. Nela são exibidas a estrutura da região juntamente com

a parte da adição de anotações(a) ilustra a estrutura do gene. (b) mostra a parte de anotação em si. O usuário pode adicionar informações textuais, com palavras-chave, qualificadores e valores, juntamente com termos de ontologia.

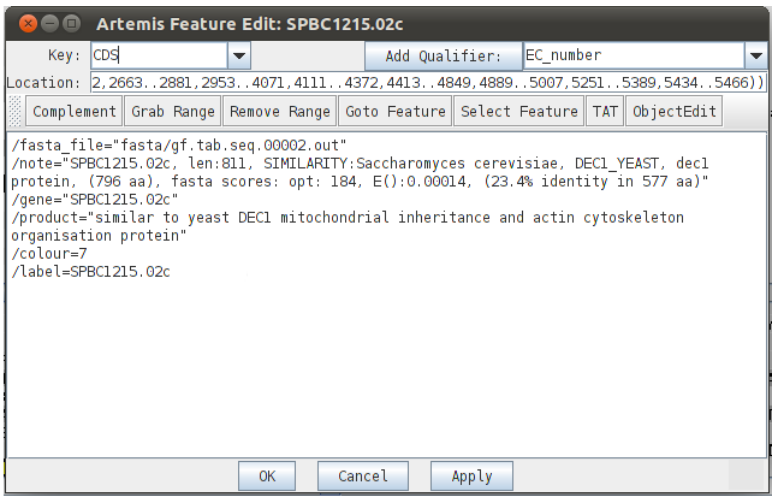


Figura 2.15: Tela secundária do *Artemis*, que ilustra o processo de anotação quando a ferramenta trabalha com manipulação de arquivos.

Apollo

O *Apollo* [LSH+02] é um editor de anotação desenvolvido pelo projeto *GMOD* com o objetivo de permitir especialistas refinarem, de forma intuitiva e flexível, as anotações das sequências geradas automaticamente por métodos computacionais. O *Apollo* permite ao usuário visualizar tanto o cromossomo inteiro bem como uma região pré-estabelecida (no *Artemis* abre apenas a sequência inteira) e oferece aos pesquisadores a capacidade de examinar, manipular e alterar a interpretação dos dados. Como entrada a ferramenta aceita diversos formatos: a partir de um servidor CGI Emsembl, arquivos no formato GAMEXML, GFF3 e outros, e conexão com banco de dados com o esquema Chado.

Diferente do *Artemis*, o *Apollo* não exibe os níveis de granularidade da sequência, pois existem apenas dois níveis de visualização: exibindo apenas o eixo (menor nível de zoom) ou exibindo os nucleotídeos da sequência (maior nível de zoom). Em relação à manipulação das características das sequências, igualmente ao *Artemis*, permite adicionar, editar e deletar características à sequência. Ajustes pontuais no nível de sequência são possíveis através do painel de editor de éxons. Em relação à adição de anotações nas características, o *Apollo* tem

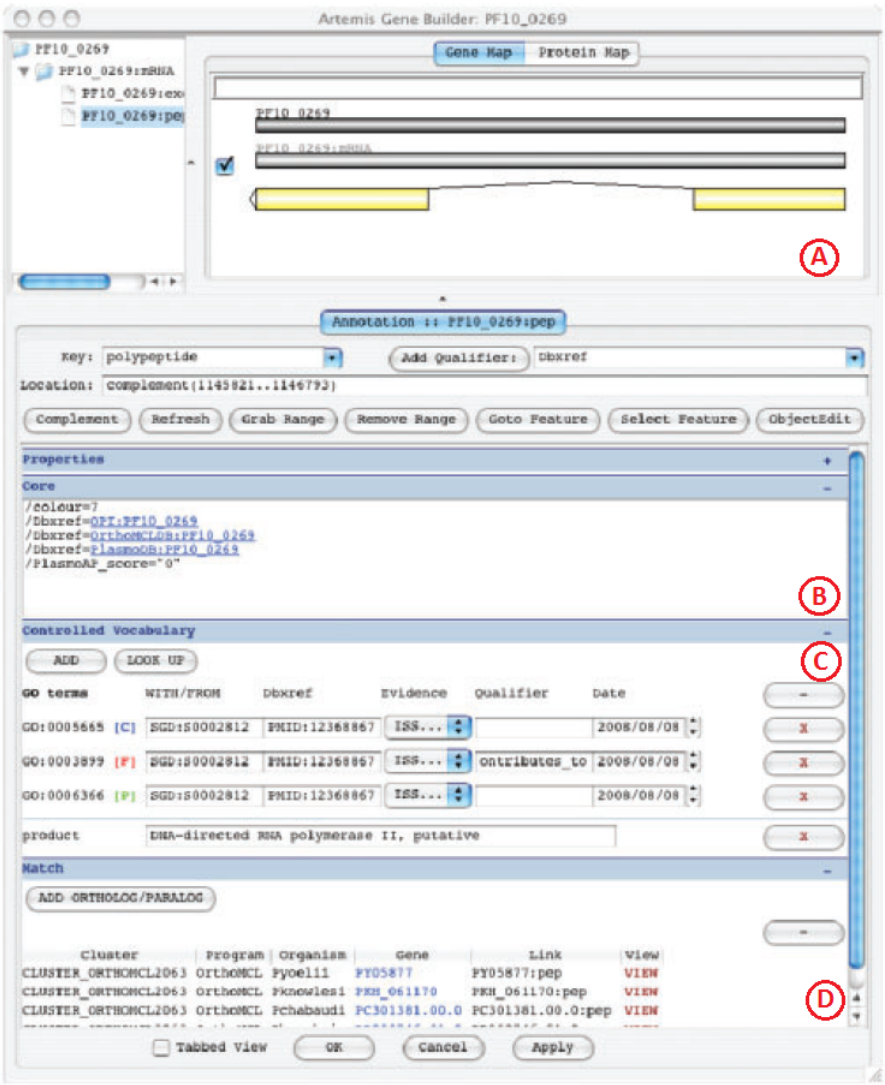


Figura 2.16: Tela secundária do *Artemis*, que ilustra o processo de anotação quando a ferramenta trabalha conectada a um banco de dados. Nela são exibidas a estrutura da região selecionada (A), o painel para a adição de informações textuais (B), a região para a atribuição de termos de ontologia à região (C) e para relacionamentos de ortologia e paralogia (D).

uma interface mais limitada quando comparado ao *Artemis*, pois permite apenas que pequenas alterações, como adição e alteração de sinônimos, adição de comentários e adição de referência da região em outra base de dados sejam realizadas e as alterações são anexadas com data e nome de quem as colocou.

Em relação à execução de análises em regiões selecionadas, o *Apollo* é mais abrangente que o *Artemis*, uma vez que disponibiliza os programas *BLAST* [SGM+90] e *Primer-BLAST* [YCZ+12] e exibe os resultados encontrados na própria ferramenta. Tais programas são executados em regiões individuais e os resultados exibidos em um painel diferenciado, que podem

se tornar características da sequência, caso seja de interesse do usuário. A figura 2.17 ilustra a interface do anotador.

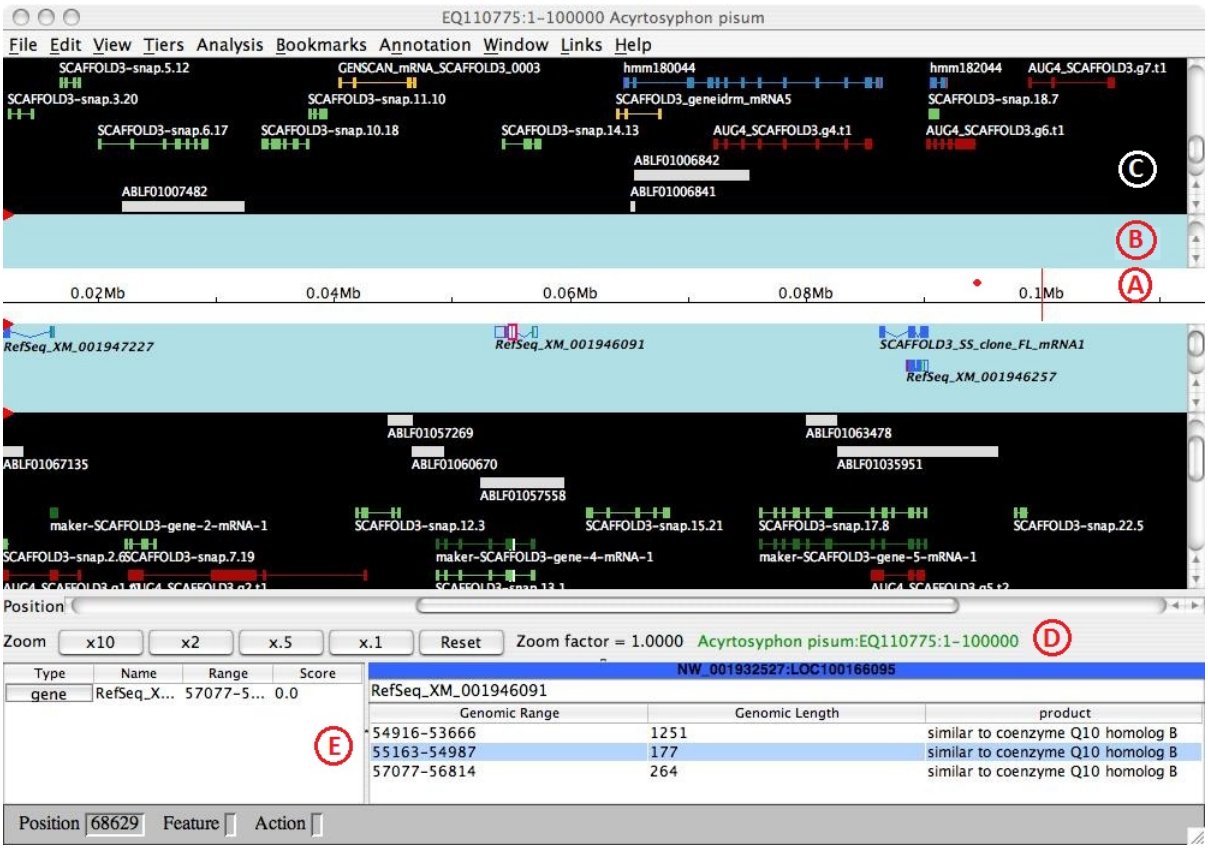


Figura 2.17: Tela principal do anotador *Apollo*. A região central (fundo branco) representa a sequência (A). Na região com fundo azul localizam-se as características da sequência (B). Na região com fundo preto, são exibidos os resultados da execução de programas (C). O usuário pode navegar na sequência a partir do painel de navegação (D). Ao selecionar uma característica, suas informações são exibidas em um painel(E).

GenDB

O *GenDB*[MGM+03] [9] consiste de um sistema desenvolvido pelo *Bioinformatics Resource Facility (BRF)* do *Center for Biotechnology (CeBiTec)* da universidade de Bielefeld, que é um *framework* extensível para anotação genômica, desenvolvido em *Perl*, para identificar, classificar e anotar genes procarióticos, usando um grande número de ferramentas. O *GenDB* possui um fluxo de trabalho permite ao usuário inserir suas sequências, processá-las, visualizar graficamente os resultados e adicionar manualmente as anotações. É importante notar que o proces-

[9] http://www.cebitec.uni-bielefeld.de/comics/index.php/gendb

samento das sequências apenas atribui a elas o que o autor denomina como observações. As anotações são atribuídas a partir de curação manual e podem seguir a ontologia *GO*, mas isso não é uma obrigatoriedade. A modelagem de dados da ferramenta é simples, com três tipos de objetos principais: região, observação e anotação, onde uma região é uma sequência ou parte de sequência a ser analisada, uma observação é o resultado do processamento e uma anotação é a interpretação da pessoa que analisa os resultados.

A figura 2.18 ilustra a tela principal do *GenDB*. Diferente do *Artemis* e do *Apollo*, as características são exibidas no painel superior e a sequência nucleotídica no painel inferior, sendo que o último exibe as região que compreende à característica selecionada.

A maior limitação dessa ferramenta é que a mesma apresenta diversos problemas para a instalação em uma máquina local, sendo que apenas funciona plenamente no ambiente criado pelos desenvolvedores da ferramenta no servidor da Cebitec. Assim, para conseguir manipular suas sequências, o usuário precisa requisitar um login e submeter suas sequências ao servidor da ferramenta, o que é uma limitação quando se trabalha com sequências que não podem ser submetidas a terceiros. Quando conectada ao servidor, o usuário pode selecionar sequências.

Discussão

Os três anotadores descritos nessa seção exibem as características da sequência em um sistema de coordenadas e permitem a anotação de forma textual livre. O *Artemis* e o *GenDB* permitem a inclusão de termos de ontologia, mas isso não é uma restrição. Quanto ao processamento de regiões específicas, apenas o *Apollo* possui ferramentas integradas para análise e que permite a inclusão dos resultados na sequência. O *Artemis* e permite a análise através de ferramenta online (o *Artemis* disponibiliza uma forma de integração com ferramentas externas, porém esta é não trivial).

2.4.5 Ferramentas integradas

As ferramentas integradas são conjuntos de programas capazes de receber sequências, processá-las a partir de *pipelines* e exibir graficamente seus resultados. Tais ferramentas podem ainda estar integradas a bancos de dados. Para ilustrar o funcionamento de uma ferramenta integrada,

Figura 2.18: Tela de anotação do *GenDB*. No painel superior (A) é possível visualizar os *contigs* da sequência, as observações, gerar relatórios e fazer buscas. O visualização atual exibe os *contigs* da sequência (B). Ao selecionar um deles, são exibidas suas características (C). Ao selecionar uma característica do *contig*, suas informações são exibidas (D) e a região do *contig* selecionado fica em destaque (E).

descreveremos três ferramentas nessa seção: *Ages* [KDC+11], *BaSys* [DSS+05] e *AutoFact* [KGLB05].

O *Ages (A Software System for Microbial Genome Sequence Annotation)* [KDC+11] integra ferramentas e bases de dados disponíveis publicamente para análise de genomas microbias. Como entrada a ferramenta aceita *contigs* em formato *FASTA* e realiza a análise a partir do seu *pipeline*, que é composto por um programa identificador de regiões de repetição (*TRF*) e por outros dois *pipelines*, *DIYA* e *PIPA*. O *DIYA* é utilizado para identificar regiões características genômicas e o *PIPA* faz a anotação funcional das regiões proteícas encontradas pelo *DIYA*. Os resultados encontrados são tanto exibidos por meio do navegador *GBrowser* como exportados em formato de arquivo de anotação *GenBank*. A figura 2.19 ilustra o fluxo de informações na ferramenta. A ferramenta está integrada a um banco de dados, o que permite que tanto os dados

de entrada como os resultados dos processamentos sejam armazenados.

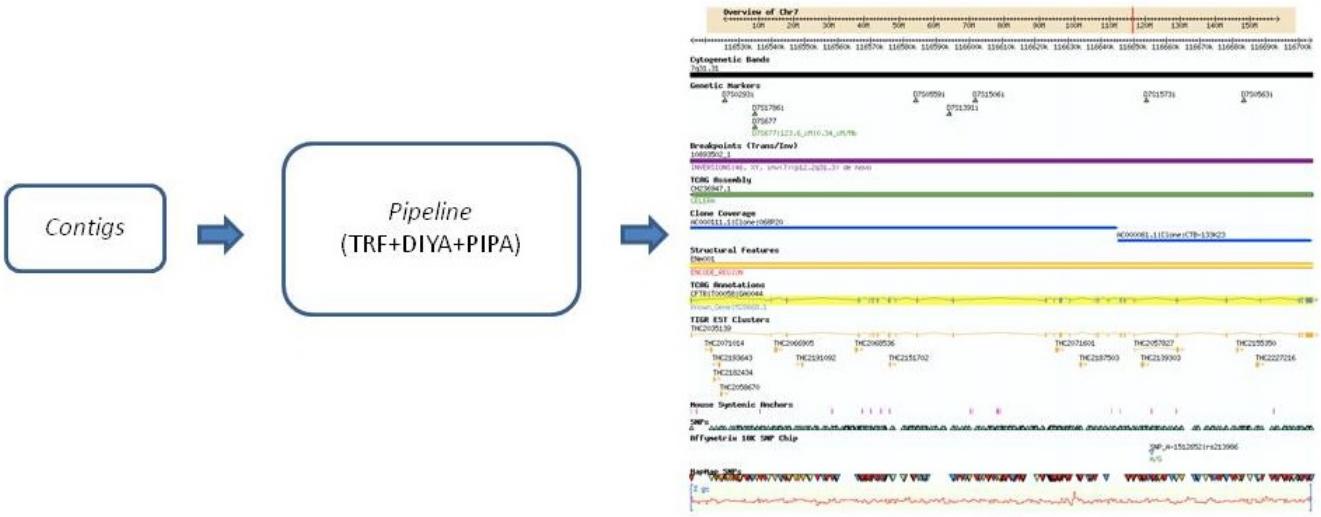


Figura 2.19: Funcionamento da ferramenta *Ages*. Os *contigs* são recebidos como entrada e processados para a identificação de suas regiões. As sequências codificadoras de proteínas são processadas pelo *pipeline PIPA* com o objetivo de atribuição de funcionalidade. Os resultados são devolvidos em formato *GenBank* e podem ser visualizados com o *GBrowser*.

O *BASys (Bacterial Annotation System)* [DSS+05] é um servidor web [10] para a anotação de genomas microbiais, que realiza a análise dos dados e exibe os resultados em um mapa de genoma circular com hiperlinks. A ferramenta é composta de três partes: (1) *interface* web para a submissão das sequências brutas, sincronização da anotação e monitoramento do processo, (2) *pipeline* para a análise dos dados e (3) geração das anotações e sistema de relatório para a exibição dos dados de saída, que podem ser páginas HTML ou arquivos de texto. O *pipeline* de anotação é composto de programas para identificação de regiões (*Glimmer* [DHK+99] e o *Critica* [BO99]) e atribuição funcional (buscas por similaridade realizadas com o *BLAST* e programas de análise de sequência). A figura 2.20 ilustra o funcionamento da ferramenta. Os dados entram na plataforma e são processados pelo *Glimmer* e pelo *Critica*. As regiões codificadores são então processadas por um procedimento de análise que integra buscas por similaridade e programas de análises de sequências.

O *AutoFact* [KGLB05] é uma ferramenta para classificação e anotação funcional de sequências. A ferramenta recebe como entra arquivos em formato fasta de nucleotídeos ou proteínas,

[10] http://basys.ca

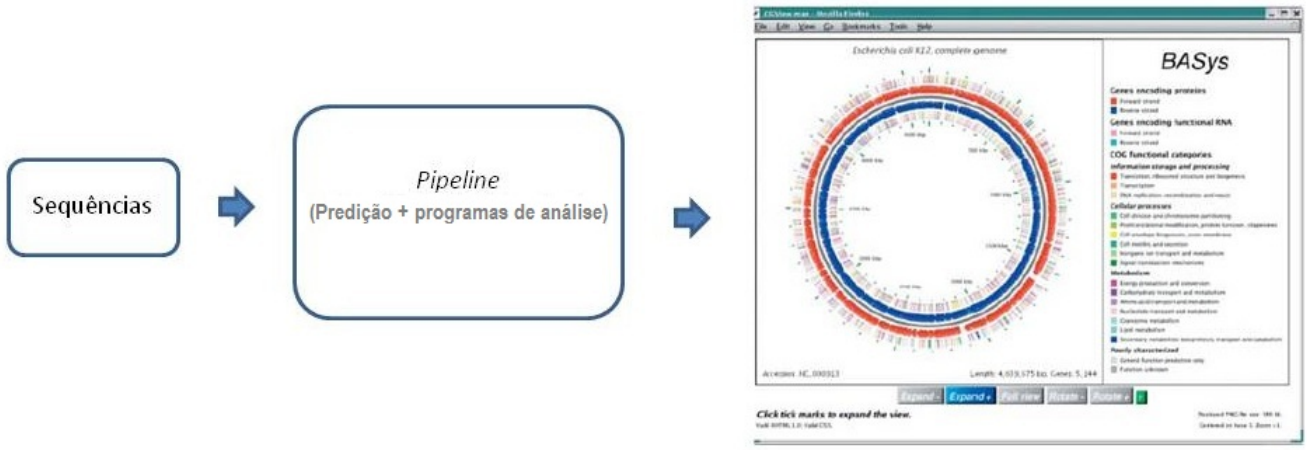


Figura 2.20: Funcionamento da ferramenta *BASys*. Os dados entram na plataforma e são processados por preditores de genes e um conjunto de programas para a busca funcional dos genes preditos. Os resultados são exibidos em páginas HTML ou em formato de arquivo.

executa uma série de pesquisas *BLAST* contra bases de dados selecionadas pelo usuário e gera arquivos de saída com os resultados, que podem ser páginas HTML, ou arquivos em formatos *GFF* ou tabular de texto. O resultados devolve a classificação das sequências, que se encaixam em uma das seis categorias: *rRNA*, proteína funcionalmente anotada, proteína não atribuída, domínio protéico, *EST* desconhecido (quando se usa dados de *EST*) e não classificado. A figura 2.21 ilustra o fluxo do sistema.

Discussão

As ferramentas descritas nessa seção compartilham duas retrições. A primeira se relaciona com o processo de análise dos resultados, uma vez que nenhuma das ferramentas descritas permite que o usuário faça anotação manual. A segunda se relaciona com as etapas fixas de processamento, pois o mesmo é realizado por um conjunto fixo de programas configurados na ferramenta.

2.5 Discussão

Como descrevemos, na literatura existem ferramentas para os diversos tipos de análise e anotação de sequências, que podem executar tarefas isoladas relacionadas a algumas das três fases do processo de anotação, como predição, busca funcional de região, visualização ou anota-

Figura 2.21: Funcionamento da ferramenta *Autofact*. As sequências entram em formato fasta, são executadas a partir de buscas de similaridade e os resultados são devolvidos em páginas HTML ou em formato de arquivo.

ção de genomas, ou fazer parte de um sistema integrado que realiza várias tarefas. Contudo, não existe uma ferramenta integrada na qual o usuário pode selecionar, processar, analizar, anotar e armazenar suas sequências, bem como recuperá-las quando necessário para novas análises. Em geral, a maioria dessas ferramentas não são gerais e se destinam a propósitos específicos, como identificar regiões de interesse e suas respectivas funcionalidades ou permitir visualização ou anotação dos resultados. A que mais se aproxima desse cenário é o *GenDB*, entretanto possui uma modelagem de dados simples, na qual apenas um subconjunto de informações resultantes de um processamento são armazenadas no banco (o artigo não informa a modelagem do banco) e a anotação é feita unicamente de forma manual (para anotação automática, é necessária a adição de *plug-ins*).

Algumas restrições são compartilhadas pela maioria dos sistemas descritos nesse capítulo. A primeira se relaciona com o conjunto de entrada para análise. Nenhuma das ferramentas existentes permite a seleção de um subconjunto de sequências. Todo o conjunto de entrada é considerado. A segunda restrição diz respeito ao tipo de organismo a ser analisado. Quase todas as ferramentas trabalham apenas com organismos procariotos, com exceção do *Genescript* e *eGene*, que não discriminam o tipo do organismo e o *PIPA* que é restrito para proteínas. Outra restrição se relaciona aos processos ou etapas fixas realizadas por essas ferramentas. A maioria delas não permite ao usuário especificar os programas a serem executados nem a ordem de execução desejada. Uma exceção a essa limitação é o *eGene*, que, por ser um sistema montador de *pipelines*, delega ao usuário a especificação dos programas e da ordem de execução, ao *GenDB*, que permite a adição de novos programas a partir de *plug-ins* e ao *DIYA*, que permite ao usuário mudar os programas da ferramenta.

A quarta limitação se refere à aceitação de dados metagenômicos. Em outras palavras, relaciona-se com a aceitação de uma grande quantidade de dados. Das citadas, apenas o procedimento operacional padrão *JCVI* foi desenvolvido com esse objetivo. As demais ferramentas não suportam tais dados, uma vez que não trabalham com tamanha quantidade de sequências, nem permitem a seleção de um subconjunto delas.

A última restrição diz respeito à atribuição de termos às regiões das sequências. Nas ferramentas que fazem a atribuição de termos às regiões da sequência, a como *Artemis*, o *Apollo*, o *Gaggle* e o *GenDB*, essa atribuição é feita livremente em formato de texto. O *Artemis*, possibilita a atribuição de termos de ontologia às regiões durante o processo de anotação, mas isso não é uma obrigatoriedade.

Capítulo 3

PATO: sistema para anotação genômica

Como vimos no capítulo anterior, existem diversas ferramentas que auxiliam nas tarefas de anotação. Essas ferramentas podem desempenhar uma tarefa específica relativa a uma das três fases do processo, como predição de regiões e análise visual dos resultados, ou integrar programas para a execução de um conjunto de tarefas. Contudo, não é possível encontrar uma ferramenta que possibilita ao usuário realizar buscas, processar, analisar, anotar suas sequências em um mesmo ambiente. As ferramentas de processamento de sequências normalmente realizam processamento e devolvem o resultado em formatos de arquivo de anotação para que possam ser analisados em ferramentas gráficas. As ferramentas integradas recebem um conjunto de dados, realizam processamentos a partir de um *pipeline* fixo e exibem os resultados por meio de um visualizador ou em páginas *HTML*, mas, não permitem a adição de informações às regiões das sequências. Finalmente, nenhuma das ferramentas analisadas disponibilizava um sistema amplo de busca e seleção de sequência baseados nos dados a ela associados. O *Artemis* pemite que o usuário faça buscas dentro da(s) sequência(s) em estudo através dos qualificadores anteriomente adicionados, porém não permite realizar busca de mais de um qualificador por vez nem de sequências no banco de dados.

Nesse contexto, o objetivo desse trabalho foi o desenvolvimento uma plataforma que permite o usuário desempenhar as tarefas de anotação em um único ambiente. A idéia é proporcionar ao usuário liberdade para trabalhar com o seu conjunto de dados, possibilitando a seleção de sequências para análise, construção dos *pipelines* processamento das mesmas e análise dos resultados a partir de um visualizador, que permite ao usuário adicionar informações às regiões

e fazer a curadoria das sequências. A possibilidade da seleção das sequências com base nas informações a ela associadas durante o processo de anotação é uma das características únicas de nosso sistema.

O desenvolvimento do sistema Pato (pipeline annotation tool) se deu através da extensão e integração da plataforma *EGene*, do visualizador *Gaggle* e do esquema de banco de dados *Chado*. Esses três sistemas foram escolhidos pela sua modularidade e facilidade de extensão.

Este capítulo descreve a plataforma desenvolvida e está dividido em duas partes. A primeira detalha o funcionamento da plataforma, descrevendo as ações possível e ilustrando o ambiente gráfico da mesma. A segunda descreve sua arquitetura, detalhando cada parte, as ferramentas integradas ao sistema e as modificações necessárias em cada uma delas.

3.1 Arquitetura

A plataforma Pato está organizada em três módulo (figura 3.1): processamento, análise e camada de dados.

O módulo de processamento é formado pelo sistema *EGene*. Sua escolha se deve ao fato de ser um sistema integrado para construção de *pipelines*, que delega ao usuário a escolha dos programas para processar as sequências. O *EGene* é composto por mais de cinquenta componentes, que abragem as tarefas de pré-processamento, coleta de evidências e geração de relatórios de anotação. A interação entre o sistema e o usuário é feita através da interface gráfica do *EGene*, o *CoEd*. O *EGene*, em sua versão original, trabalhava apenas com a manipulação de arquivos e gerava como resultado dos processamentos arquivos XML. Contudo, o uso de um arquivo limitaria a plataforma em termos de número de sequência e alternativas de busca. Desta maneira decidimos extender a plataforma *EGene* para uso de um banco de dados, sendo escolhido para isso o esquema *Chado*.

O módulo de análise consiste de um ambiente gráfico responsável pela comunicação com o usuário. Todas as ações (com exceção da montagem dos *pipelines*), desde a inserção as sequências até a atribuição de conclusões às suas regiões, são realizadas através desse componente. Esse módulo consiste de um conjunto de telas que se comunicam com a base de dados e com o módulo de processamento a fim de executar ações para a análise dos dados.

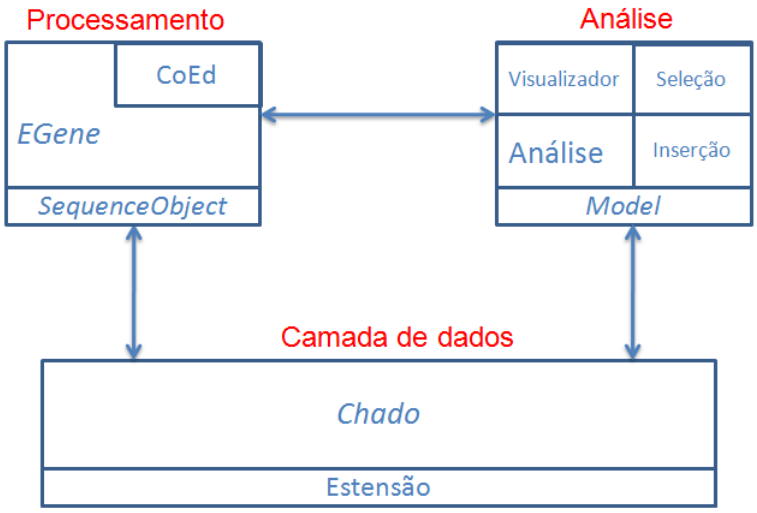


Figura 3.1: Arquitetura da plataforma proposta. O módulo de processamento é composto pelo *EGene* e por sua ferramenta gráfica. O módulo de análise é responsável pela comunicação com o usuário. Todas as ações da plataforma (com exceção da criação de *pipelines*) são executadas a partir dele. A base de dados contém o esquema *Chado*, que foi extendido para atender as necessidades da plataforma. Os módulos se comunicam com a base de dados através de suas camadas de representação.

3.1.1 Extendendo o modelo de dados do *EGene*

O *EGene* é um sistema construtor de *pipelines* para processamento de sequências. Seus componentes abrangem as tarefas de pré-processamento, coleta de evidências e geração de relatórios. Os componentes de pré-processamento são responsáveis por preparar as sequências brutas (resultantes de um processo de sequenciamento) para a etapa de coleta de evidências. Operações como trimagem, mascaramento e invalidação se relacionam com esse processo. A trimagem consiste na invalidação de uma parte das extremidades da sequência. O mascaramento consiste na substituição dos resíduos da sequência pela letra *x*, que indica que um conjunto de posições consecutivas da sequência que devem ser desconsideradas. As operações de invalidação tornam as sequências obsoletas por algum motivo, como baixa qualidade ou tamanho. Sequências inválidas são ignoradas pela maioria dos componentes, mas são mantidas no *pipeline*.

A coleta de evidências é responsável pela busca de informações na sequência. Nesse processo, os componentes executados devolvem como resultado evidências, que são regiões da sequência identificadas por algum processo. Os componentes de geração de relatórios devol-

vem os resultados dos processamentos em formatos de arquivo de anotação.

A fim de armazenar as informações a respeito dos processamentos realizados nas sequências, a camada de representação do *EGene* modela quatro tipos de dados: sequência, *log*, operações e evidência. A sequência é um registro que armazena os resíduos, o vetor de qualidade (associa um valor numérico a cada posição da sequência), o nome e o tipo da sequência. Um *log* é um registro que representa um processamento. As informações armazenadas em um *log* são nome e versão do componente utilizado, data do processamento, nome e versão dos programas que fazem parte do componente. As operações podem ser de três tipos: trimagem, mascaramento e invalidação. Cada um dos registros de operação está associado a um *log*.

Por fim, uma evidência representa a informação gerada por um *software* que analisa a sequência. Cada evidência está associada a um *log*. As evidências são classificadas em quatro tipos:

- Multirregião: Consiste de uma ou mais subsequências da sequência original. Cada subsequência pode estar associada a diferentes *tags* e valores numéricos.

- Similaridade: Representa o resultado do alinhamento de determinada região da sequência com sequências de base de dados externas. Cada entrada de similaridade descreve as coordenadas do alinhamento, o banco de dados no qual a sequência alinhada está contida, um conjunto de valores numéricos e uma *tag* em formato de texto que representa o alinhamento.

- Estatística: Representa uma informação tabular a certa de uma região, como por exemplo tabela de uso de códons e frequência de cada base.

- Gráfica: Atribui um valor numérico para cada base da região e pode ser representada como um gráfico de duas dimensões. Um exemplo desse tipo de dado é o conteúdo *GC* das sequências.

No Pato, além dessas informações, é preciso registrar as informações que se relacionam com o processo final de anotação, que são as conclusões. Essas informações são adicionadas a partir de termos de ontologia e podem ou não se basear em evidências. A fim de permitir

que conclusões sejam adicionadas às sequências, extendemos o modelo do *EGene* para englobar esse tipo de característica. Informações como tipo (automática ou manual), usuário que a criou (se for automática, será atribuído o nome do programa) e se é curada (analisada por um especialista) ou não são registradas. É permitido inclusive adicionar informações adicionais, que são anexadas na forma de comentários. De forma similar às evidências, as conclusões estão igualmente associadas a um *log*.

3.1.2 Camada de representação

Para modelagem dos dados no *Pato* foi escolhido o esquema *Chado* [MEC07]. O determinante nessa escolha foi a sua arquitetura genérica que, permitiu um mapeamento mais natural da maior parte do modelo do *EGene*. Porém, para garantir a representação de todas as informações geradas pelo sistema, foi necessário extender o esquema *Chado*. Nessa seção descreveremos o mapeamento dos tipos de dado do *EGene* (sequência, *log*, trimagem, mascaramento, invalidação, evidência e conclusão) no *Chado*.

Para a representação de sequência, evidência e conclusão utilizamos o módulo *Sequence* (figura 3.2) do *Chado*. Foi preciso extender uma de suas tabelas (*feature_cvterm*) para representar todas as informações relativas às conclusões. Os *logs* do *EGene* foram mapeados no módulo *Companalysis* (figura 3.4), que também foi extendido para possibilitar o uso de outros *softwares* por parte dos componentes do *EGene*. Finalmente, para a representação das operações foi necessária a criação de um novo módulo (figura 3.6). Em seguida descreveremos os mapeamentos e as modificações realizadas.

Módulo *Sequence*

O *Chado* é um esquema baseado em sequência e tem o módulo *Sequence* como central. Para o *Chado*, toda sequência (ou parte dela) é caracterizada como uma *feature*, que é definida como uma região de uma macromolécula biológica (*DNA, RNA* ou polipeptídeo) ou um agregado de regiões desse polímero. No esquema, as *features* são tipificadas a partir de termos de ontologia. Conforme o conceito de *feature*, uma sequência pode ser mapeada como uma *feature*. Como evidências e conclusões são regiões da sequência original, essas são igualmente classificadas

como *features*. Assim, sequências, evidências e conclusões são armazenadas na tabela *feature* (tabela principal do módulo *Sequence*) e são diferenciadas a partir dos tipos, atribuídos por termos de ontologia.

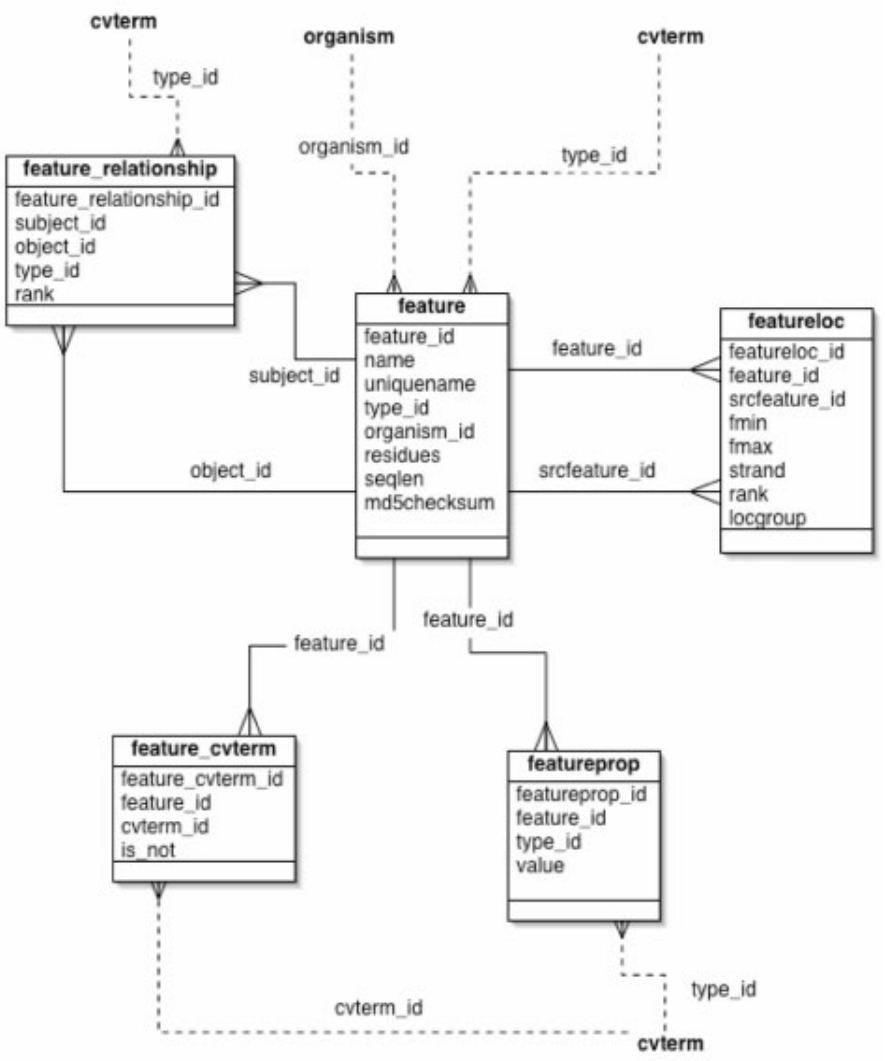


Figura 3.2: Esquema do módulo *Sequence* do esquema *Chado*. A tabela *feature* é a tabela central do esquema. A tabela *feature_relationship* armazena os relacionamentos entre as *features*, *featureloc* localiza uma *feature* em uma referência e a *featureprop* armazena as propriedades das sequências. As informações atribuídas através de termos de ontologia são armazenados na tabela *feature_cvterm*.

As demais tabelas do módulo *Sequence* representam o relacionamento entre duas *features* (*tabela feature_relationship*), a localização de uma *feature* em relação a outra (*featureloc*), as propriedades (tabela *featureprop*) e os termos de ontologia atribuídos a uma *feature* (tabela *feature_cvterm*). Utilizando as demais tabelas do esquema, conseguimos mapear o relacionamento entre sequências, evidências e conclusões, bem com a localização das evidências e conclusões

na sequência referência.

As conclusões recebem termos de ontologia a fim de descrever uma região. Como a tabela *feature_cvterm* faz a associação de termos de ontologia a uma *feature*, os termos atribuídos a uma conclusão são armazenados nessa tabela. Contudo, a tabela *feature_cvterm* não é capaz de armazenar todas as informações a respeito da conclusão (tipo, usuário e se a conclusão é curada). Adicionamos os campos *type*, *user*, *curated* (que se relacionam respectivamente com o tipo da conclusão, usuário que criou a criou e se a conclusão é curada) à tabela *feature_cvterm*, de forma a todas as armazenar as informações a respeito de uma conclusão. Os comentários de uma conclusão são associados à conclusão como propriedades, a partir da tabela *featureprop*. O tipo dos relacionamentos e as propriedades são definidos por termos de ontologia. A figura 3.3 mostra um exemplo de como sequências, evidências e conclusões são armazenadas no *Chado*.

Módulo *Companalysis*

O esquema *Chado* possui um módulo, o *Companalysis*, que trata das análises realizadas nas sequências. O conceito fundamental desse módulo é que os resultados de uma análise computacional podem ser interpretados ou descritos como uma *feature*. Na nossa plataforma, os processamentos geram evidências ou conclusões, que são *features*. Assim, o *log* foi modelado como uma análise no *Chado*. Os atributos do *log* foram mapeados como atributos da análise, sem perda de informação. Ao observar os *logs* gerados pelo *EGene*, observamos casos em que componentes fazem uso de programas auxiliares. Como o módulo *Companalysis* não possui recursos para definir relacionamentos entre análises, extendemos o módulo através da criação da tabela que armazena os relacionamentos entre duas análises e define o tipo do relacionamento por meio de um termo de ontologia. A figura 3.4 ilustra as tabelas do módulo.

O módulo *Companalysis* possui uma tabela que relaciona as análises e as *features* resultantes do processo (tabela *analysisfeature*). Essa tabela foi utilizada para relacionar os *logs* e as evidências e conclusões geradas por um componente. A figura ilustra o mapeamento de um *log* e suas evidências no *Chado*. Ao processar uma sequência, cada componente do *pipeline* gera um *log*, que registra informações a respeito do componente executado. Caso evidências sejam encontradas, essas se relacionam com o componente a partir do *log*.

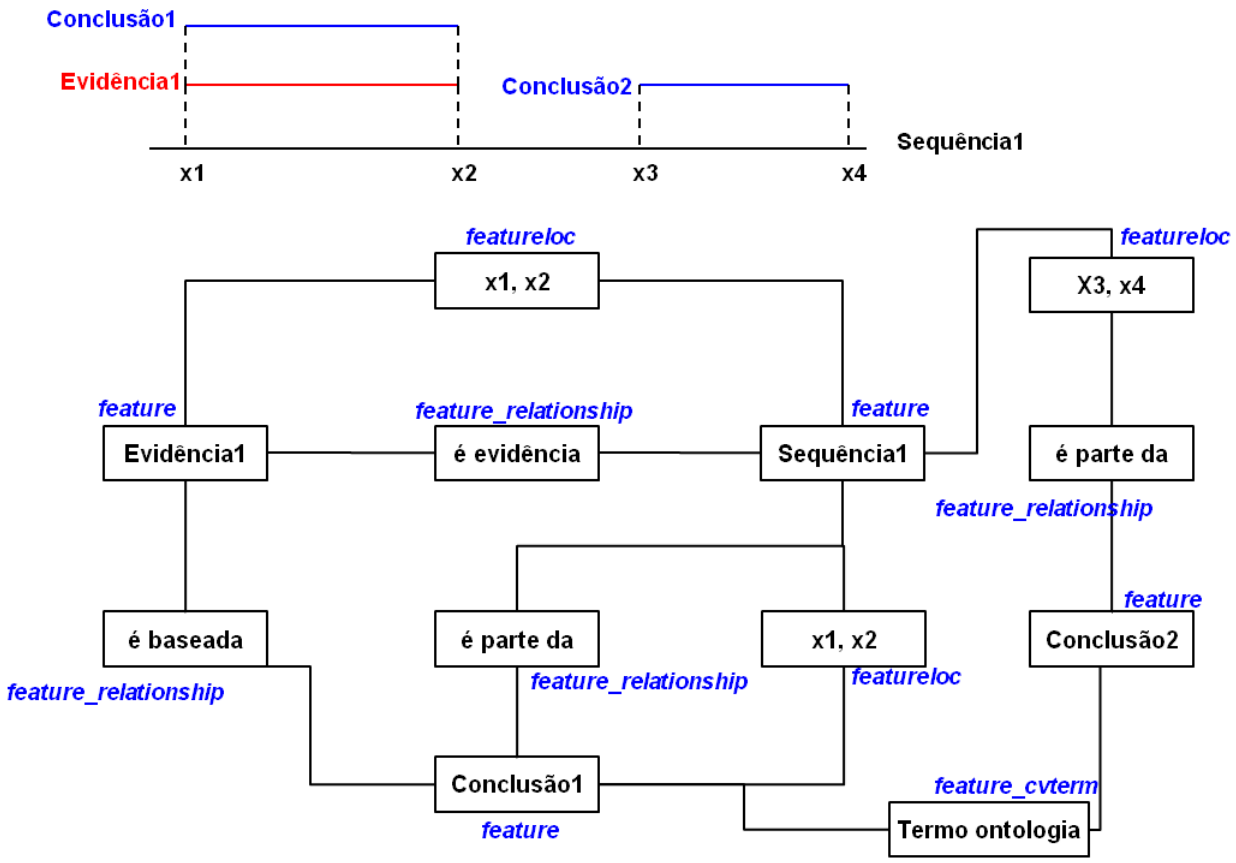


Figura 3.3: Na figura é possível ver a representação de uma evidência e duas conclusões em relação à uma sequência. As três características possuem relacionamentos e se localizam na sequência. Uma das conclusões (*Conclusão1*) possui um relacionamento com evidência, pois foi criada a partir da evidência. Mapeando tais informações no *Chado*, temos que a *Evidência1* é uma evidência da *Sequência1* e que as conclusões são parte da sequência. Os relacionamentos são armazenados na tabela *feature_relationship* e identificados por termos de ontologia. O relacionamento entre a evidência e a conclusão é igualmente armazenado nessa tabela. As três características localizam-se na sequência através de coordenadas e essas informações são armazenadas na tabela *featureloc*. Por fim, as conclusões recebem termos de ontologia, que são armazenados em *feature_cvterm*.

Módulo *Pre_processing*

As operações (trimagem, mascaramento e invalidação) são ações que não geram evidências, apenas realizam modificações na sequência original. Dessa forma, não há como armazenar as informações das informações no módulo *Companalysis*. Para o armazenamento dessas informações, criamos o módulo *Pre_processing*, que é composto por três tabelas, uma para cada tipo de operação, e que associa um *log* a cada uma dessas operações. A figura 3.6 ilustra o módulo

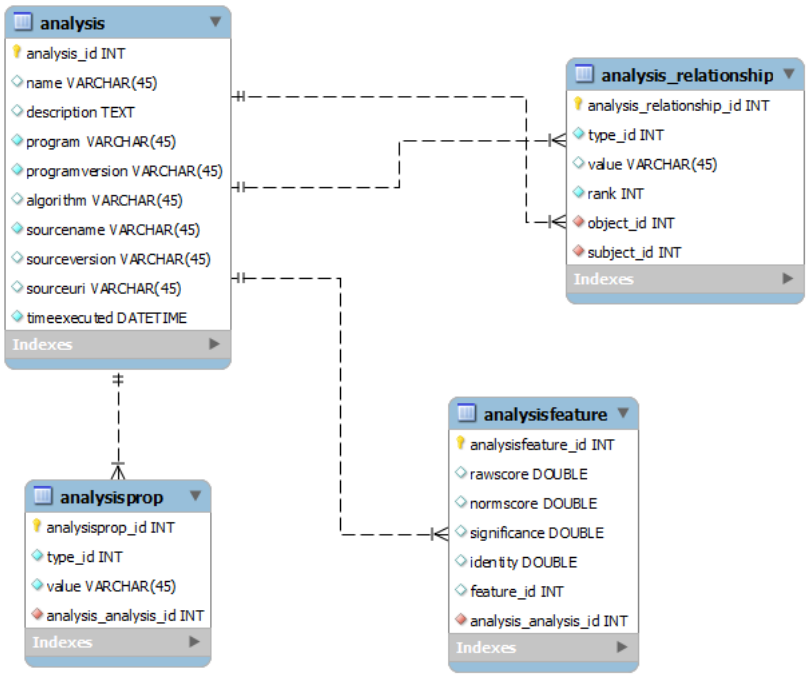


Figura 3.4: Esquema do módulo *Companalysis* com a adição da tabela *analysis_relationship*.

Pre_processing.

Módulo *Controlled Vocabulary*

Como vimos, o esquema *Chado* utiliza termos de ontologia não somente para anotar as sequências, mas também para tipificar sequências, relacionamentos e propriedades. O módulo responsável pelo armazenamento dessas ontologias e vocabulários controlados é o *Controlled Vocabulary*. Quando o *Chado* é instalado, algumas ontologias são carregadas, como a ontologia de sequência (*SO*) e a de gene (*GO*). Contudo, o *EGene* faz uso de termos específicos para classificar seus resultados, que não estão presentes em nenhum dos vocabulários do banco. A fim de suprir a falta desses termos, criamos um vocabulário controlado próprio, denominado de *egene_cv*, que contém os termos específicos do *EGene* necessários para a identificar o tipo das sequências, propriedades, anotações e relacionamentos utilizados na plataforma.

Os demais módulos utilizados e extendidos são descritos no apêndice desse documento.

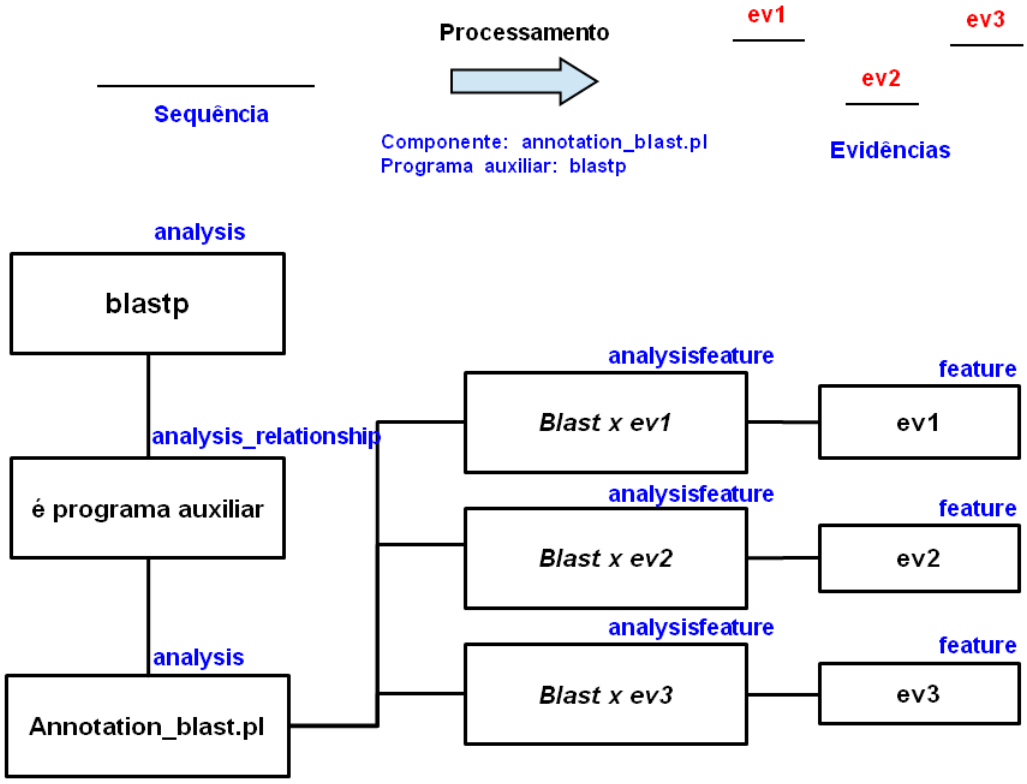


Figura 3.5: Modelagem de uma análise do *EGene* no módulo *Companalysis*.

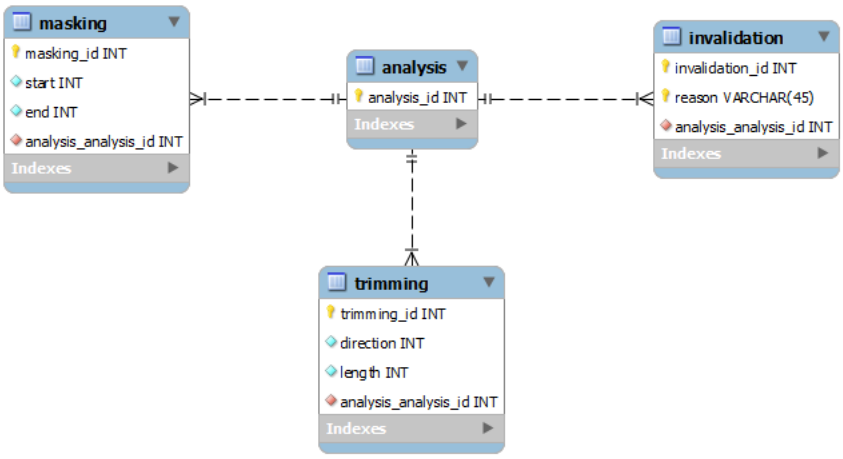


Figura 3.6: Módulo criado para o armazenamento das operações de pré-processamento. Cada operação se relaciona com um *log* e, por isso, há o relacionamento de cada uma das tabelas com a tabela de análise.

3.1.3 Módulo de processamento

O módulo de processamento é composto pelo sistema contrutor de *pipelines EGene* [DKM+05]. O *EGene* possui uma arquitetura modular formada por três partes: *pipeline runner*, componentes de processamento e camada de representação. O *pipeline runner* é responsável por iniciar cada processo do *pipeline* e estabelecer a comunicação entre seus componentes.

Os componentes do *EGene*, como descrevemos no capítulo dois, são *scripts Perl* que processam as sequências. O sistema proporciona uma completa liberdade para a implementação interna dos componentes, porém há um padrão para a formatação de suas entradas e saídas. Cada componente lê um "objeto de sequência"como entrada e produz um "objeto de sequência"com os resultados encontrados no processamento adicionados ao objeto de entrada, utilizando a camada de representação.

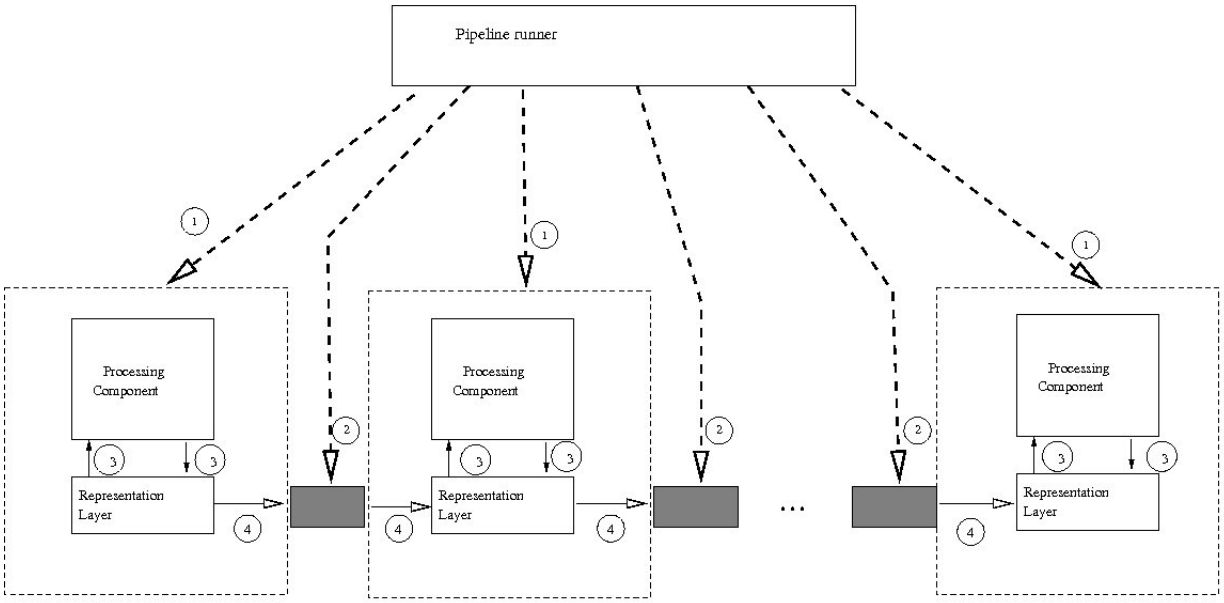


Figura 3.7: Arquitetura do sistema *EGene*. Um *pipeline* é composto de vários processos independentes, um para cada componente. O *pipeline runner* inicia os processos individuais (1) e os canais de comunicação (2). Cada processo é composto de um *script* de análise e uma instância da camada de representação, utilizada para atualizar os dados (3). Cada processo gera entradas e saídas em formato padrão (4). Figura extraída de [FAN+nd].

3.1.4 Componente de análise

O componente de análise consiste de um ambiente gráfico, desenvolvido em *Java*, por meio do qual o usuário desempenha as tarefas da plataforma. A partir dela o usuário pode inserir sequências, selecionar um subconjunto para análise, invocar o módulo de processamento, analisar visualmente os resultados e atribuir conclusões.

Para o desenvolvimento desse ambiente, foi utilizado o padrão *MVC (Model-View-Controller)* [EHJV00]. No sistema, o objeto modelo é representado por um pacote de classes, denominado *Model*, que contém as classes que se comunicam diretamente com a base de dados para a troca de informações. Essas classes representam os objetos utilizados na plataforma, como sequência, evidência, processamento e conclusão (respectivamente *Sequence*, *Evidence*, *Analysis* e *Conclusion*). O pacote *Controller* representa o conjunto de classes que faz o intercâmbio de informações entre o banco e a interface do usuário. Por fim, a Visão da plataforma é representado pelo pacote *UI*, que contém as classes desenvolvidas para a interface visual da ferramenta.

O módulo de análise se comunica com o de processamento através da execução de *pipelines*, que podem ser criados pelo usuário ou pela plataforma. O usuário cria os *pipelines* a partir da ferramenta gráfica do *EGene*, o *CoEd*. A plataforma monta *pipelines* de forma transparente ao usuário. A execução os *pipelines* é iniciada pelo componente de análise.

O primeiro passo para realização análises a partir da plataforma é a inserção das sequências no banco. Esse procedimento é realizado através da integração do componente de análise com o *EGene*. O sistema *EGene* possui componentes que executam a tarefa de inserção de sequências em alguns formatos (no caso da plataforma, os formatos de interesse são o fasta e o xml). Para a execução desse processo, a interface monta um *pipeline* e executa o seu processamento. Ao terminar, o módulo de processamento retorna os identificadores das sequências, permitindo a continuação da análise.

Um característica-chave da plataforma é a possibilidade de seleção de um subconjunto de sequências. Essa seleção é feita através de parâmetro escolhidos pelo usuário. Tais parâmetros foram definidos em parceria com um biólogo, de forma a tornar as consultas úteis do ponto de vista do usuário. Para cada conjunto de parâmetros selecionados, uma consulta *SQL* é formulada indepentende da interferência do usuário. A consulta é executada e os resultados são retornados.

O *Gaggle Genome Browser (GGB))* [BKR+10] foi escolhido como base do sistema de visualização do Pato por ser de código aberto, extensível e possuir um modelo de dados compartilhado por diversos pacotes de *software* de genômica. Neste modelo existem quatros elementos principais: *Sequence*, *features*, *tracks* e *dataset*. *Sequence* é a unidade básica do esquema que representa as coordenadas e resíduos da sequência; *features* representam regiões da sequência as quais são associadas algum tipo de informação; *tracks* agrupam as *features* de uma fonte comum e os *dataset* que agrega as demais informações.

As *features* da sequência são desenhadas no visualizador a partir da implementação da classe abstrata *TrackRenderer*. A visualização das *features* são implementadas pela extensão dessa classe, que codificam visualmente suas propriedades utilizando cores, formatos e outros recursos.

Para a adequação do visualizador à plataforma foram realizadas algumas alterações. A primeira se relaciona com a conexão com um banco de dados. Em sua versão original, o *GGB* trabalha apenas com manipulação de arquivos. Foi preciso, então, contectar o navegador ao nosso banco de dados. Para tanto, criamos uma classe que recupera a sequência e suas características do banco e monta um *dataset*. Este *dataset* é então repassado para a aplicação para a exibição das sequências na ferramenta.

O *GGB* é capaz de exibir diversos tipos de dados. Porém, em sua versão original, não existiam tipos capazes de representar as evidências e conclusões do *EGene*. Assim, foi preciso criar *features* com essa finalidade. Implementamos três tipos de *features*: um para representar as evidências de multirregião, similaridade e estatísticas (compreendem regiões da sequência que possuem informações relacionadas), um para as evidências gráficas (atribuem um valor para cada posição da região e são representadas como gráficos) e um para a representação das conclusões. Para a representação visual dessas *features*, foram criadas três classes que extendem a classe *TrackRenderer*. A figura 3.8 ilustra o modelo do *Gaggle* extendido para a plataforma. A abstração principal está em azul e em verde estão as classes responsáveis por desenhar as *features* na tela.

Todas as evidências, com exceção das gráficas, são plotadas como retângulos, de diferentes cores. Cada tipo de evidência é desenhado em um nível do painel: as de multirregião estão

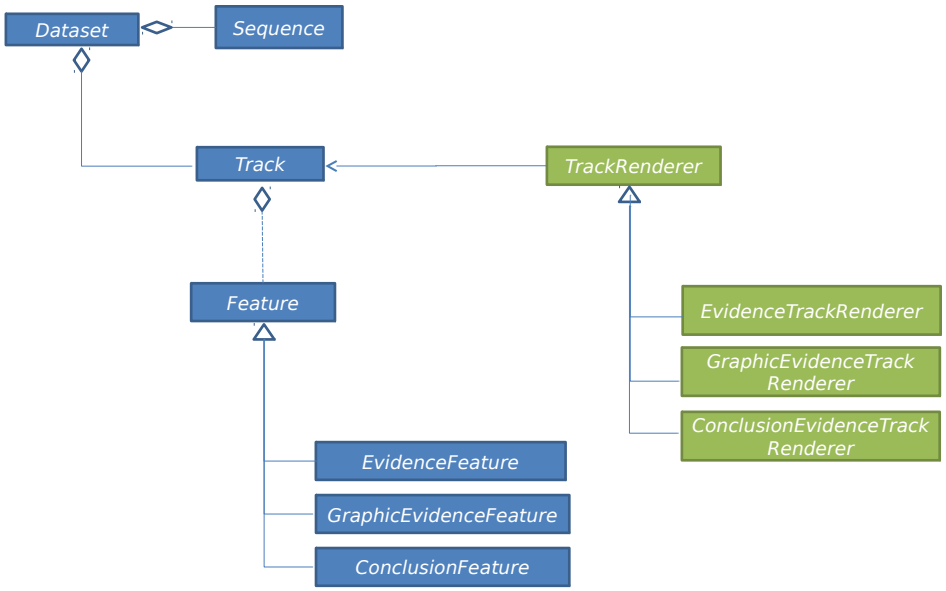


Figura 3.8: Modelo de dados o *GGB* extendido para a plataforma.

mais próximas ao eixo de coordenadas, seguidas das de similaridade, estatísticas e gráficas. As evidências podem ter subevidências (evidências encontradas a partir do processamento de uma evidências). As subevidências são desenhadas imediatamente abaixo ou acima da evidência a depender da fita da sequência (fitas direta e reversa respectivamente) e sua cor é atribuída de acordo com seu tipo. Para diferenciar as evidências principais (encontradas a partir da sequência original) das subevidências, as evidências são desenhadas em dois tons, enquanto as subevidências são desenhadas com cores sólidas. As evidências gráficas utilizaram o padrão de gráfico do *GGB*.

As conclusões desenhadas como retângulos em um painel distinto das evidências. Para cada seu tipo (manual ou automática) foi atribuída uma cor. A fim de destinguir as curadas das não curadas, as conclusões que já analisadas recebem uma tonalidade mais escura.

Em relação às características visuais, originalmente o *GGB* é composto de apenas um painel que exibe as características da sequência, como ilustrado na figura 3.9. A fim de separar as evidências das conclusões, foi adicionado um outro painel, abaixo e no mesmo formato do anterior, para a exibição das conclusões. Um terceiro painel foi incluído à interface, com o objetivo de exibir as análises realizadas na sequência. Tais análises são exibidas por meio de

uma árvore, onde os nós representam os programas executados e as folhas as evidências encontradas. Finalmente, foram realizadas pequenas alterações na interface: inserção de *Listbox* para listar as sequências selecionadas pelo usuário. Na plataforma, o usuário pode visualizar a sequência inteira ou apenas uma região específica dela. Como essa característica não estava implementada, foram adicionados *Checkboxs* para qua o usuário informe o como pretende visualizar a sequência e caixas de textos para a indicação das coordenadas que deseja visualizar (caso a visualização seja parcial). A figura 3.10 ilustra a tela atual do sistema.

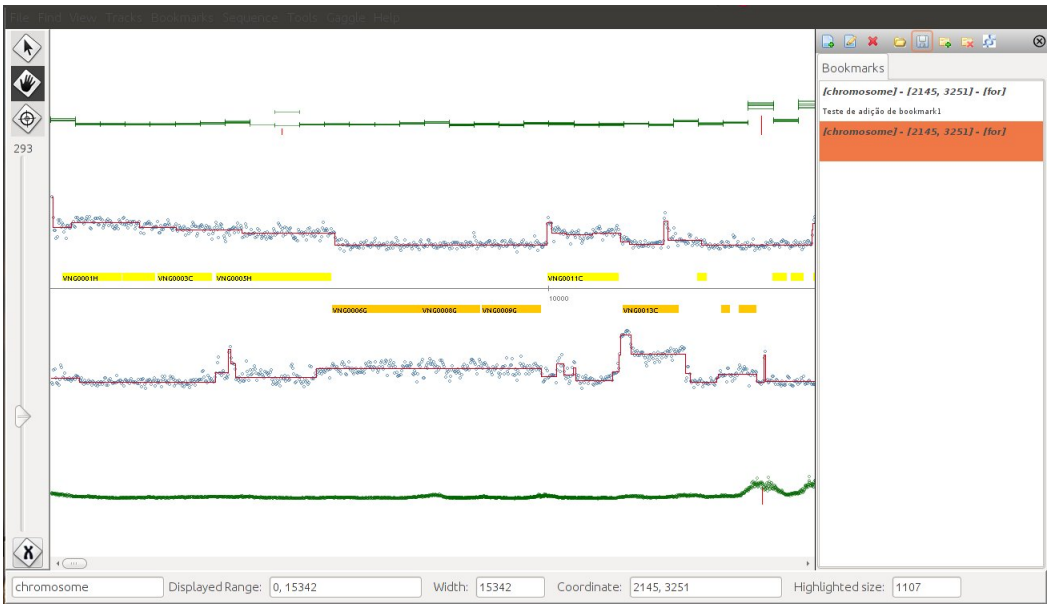


Figura 3.9: Tela original do visualizador *Gaggle*.

Para uma característica selecionada, o *Gaggle* disponibilizava poucas ações, como exibir informações, modificar propriedades visuais (como cor ou se devem ficar visíveis) ou adicionar um informações à região. Na versão extendida foram adicionadas algumas opções de análise. Para as evidências selecionadas, além das propriedades da evidência, é possível visualizar sua sequência de nucleotídeos, de aminoácidos (caso seja uma região codificante de proteína) e é permitido realizar processamento, seja por meios de ferramentas online (para pequenas verificações) ou através do módulo de processamento.

Na versão original da ferramenta, as informações adicionadas a uma região eram feitas em formato de texto, a partir da adição de *bookmarks*. Na plataforma essa característica foi desconsiderada. Agora o usuário atribui conclusões às regiões das sequências por meio de

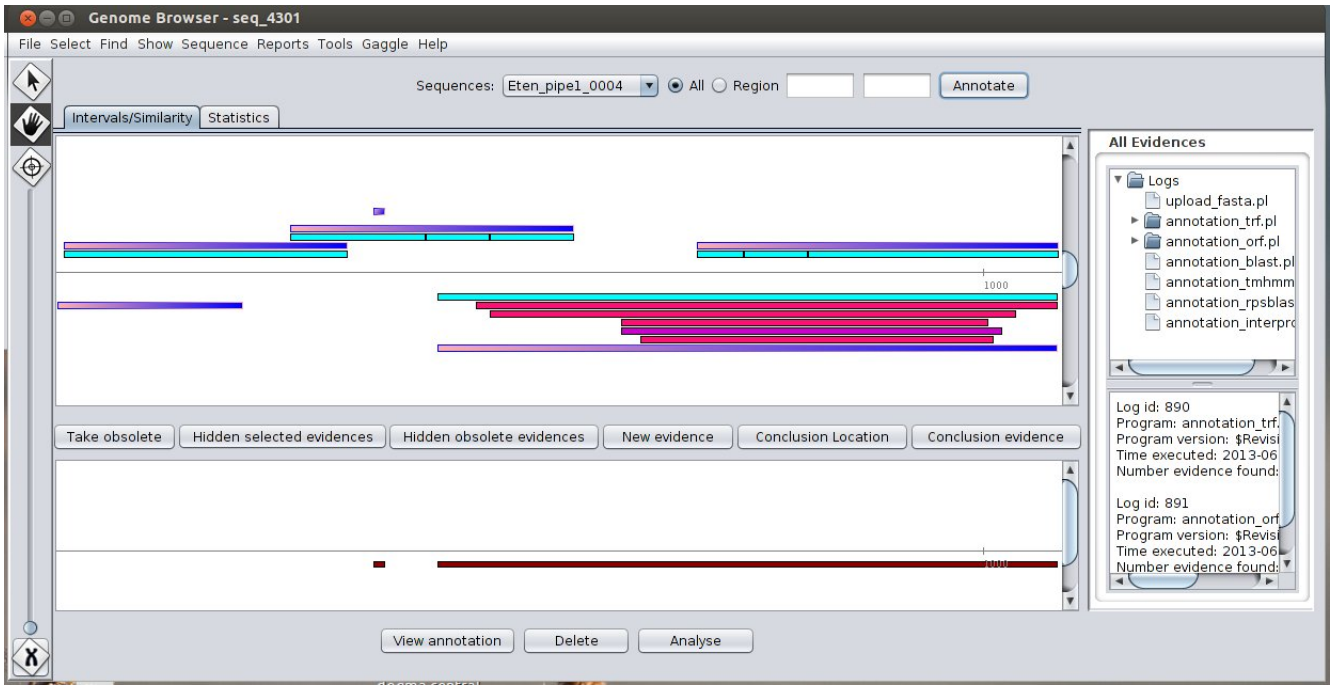


Figura 3.10: Tela modificada do *Gaggle*. Foram adicionados dois painéis, um para a exibição das conclusão e outro para as análises realizadas na sequência. As evidências são ilustradas no primeiro painel, sendo que as evidências principais são desenhadas de duas cores e as subevidências em cores sólidas. O painel inferior exibe as conclusões. O painel lateral direito ilustra as análises executadas na sequência.

termos de ontologias que as descrevem ou associam funcionalidade.

3.2 Funcionalidades da plataforma

O Pato permite que o usuário inicie o processo de anotação tanto a partir de um arquivo contendo um conjunto de sequências quanto a partir de uma consulta ao banco de dados. A partir daí, um conjunto de interfaces gráficas guia o usuário no processo de visualizar as informações da sequência, anexar conclusões, indicar quais informações associadas a uma sequência são relevantes e fazer o refinamento das buscas já realizadas. Novas informações podem ser adicionadas às sequências de maneira automática através da interface gráfica do EGene e da geração e conjuntos de indentificadores feitos a partir da interface do Pato.

A figura 3.11 ilustra em mais detalhes os possíveis fluxos de uso do sistema. Inicialmente, o sistema permite que o usuário insira novas sequências a partir de arquivos ou as selecione do banco de dados, destacados com (1) na figura. Com as sequências selecionadas (2), o usuário

pode visualizar de forma geral uma sequência, processar o conjunto de dados ou analisar uma sequência individualmente (3). Caso a opção seja a de analisar uma sequência individualmente (4), a sequências e suas características serão exibidas graficamente para o usuário. A partir de então o usuário pode selecionar uma evidência (5), atribuir uma conclusão (6) ou selecionar uma conclusão (7). Caso uma evidência seja selecionada, as opções como visualizar suas propriedades e resíduos, processar ou tornar a evidência obsoleta ou invisível serão disponibilizadas. Caso o usuário selecione uma conclusão, as ações de exibição das propriedades, edição, deleção e curagem estarão disponíveis.

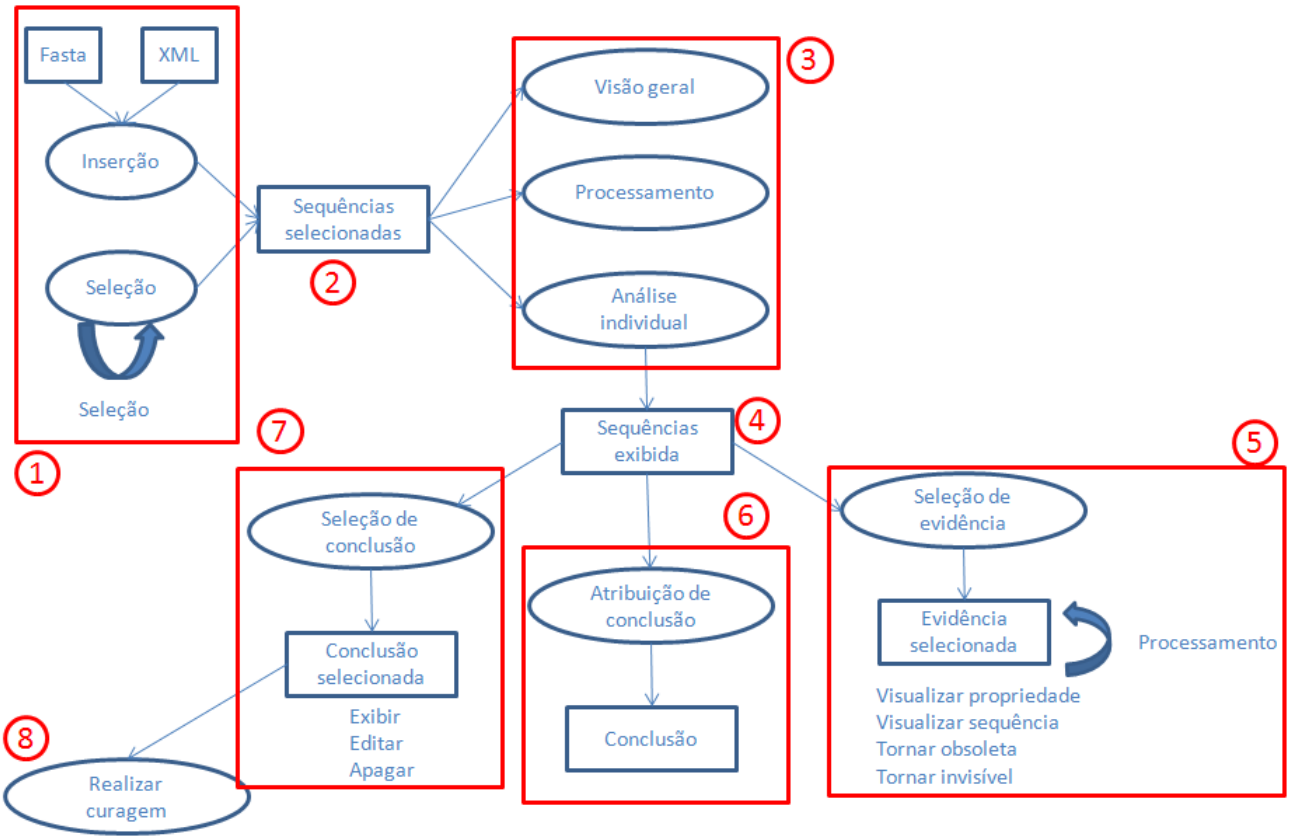


Figura 3.11: Fluxo de uso da plataforma.

3.2.1 Inserção de novas sequências na plataforma

O processo de inserção através da plataforma *EGene*, que possui um conjunto de componentes para a entrada de sequências no sistema. Assim, os tipos de arquivos aceitos pela plataforma são os mesmos aceitos pelo sistema *EGene*, que atualmente são o fasta (sequências brutas) e

o XML do *EGene* (sequências previamente processadas). No Pato, para a inserção de sequências em formato fasta, o usuário deve informar um nome para as sequências caso não queira utilizar o cabeçalho de cada sequência (renomeá-las a partir de um prefixo), o projeto de sequenciamento que a originou e o organismo ao qual pertence. O organismo é definido através da seleção de um dos organismos listados pelo sistema. Caso a espécie em estudo não esteja contida no banco de dados, o usuário é capaz de inserí-la. Para arquivos em formato *xml* do *EGene*, as informações são similares, exceto que não é preciso informar o nome nem a composição das sequências (para a inserção de dados no formato xml não é possível modificar o nome da sequência). A figura 3.12 ilustra as telas de inserção de sequências e de um organismo. As sequências inseridas no sistema são consideradas selecionadas.

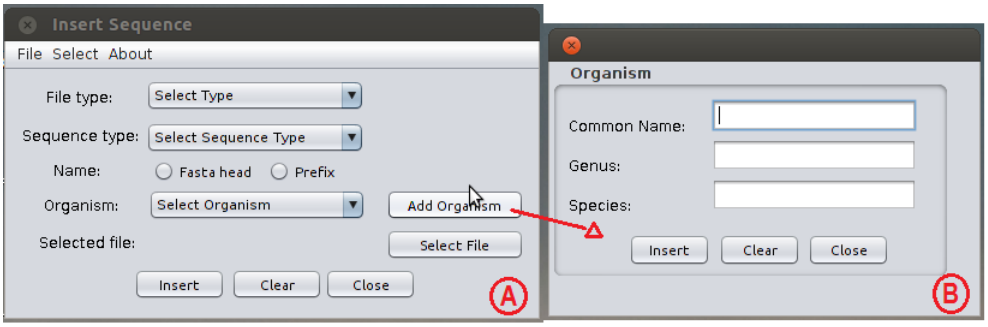


Figura 3.12: Tela de inserção das sequências (A). O usuário primeiramente seleciona o tipo do arquivo. Caso seu arquivo esteja em formato fasta, o usuário irá preencher todos os campos da tela. Caso seja xml, os campos referentes ao tipo da sequência e nome ficam desabilitados. Ao selecionar o organismo, caso o desejado não esteja no banco, é possível inserí-lo (B).

3.2.2 Seleção de sequências

As sequências do banco de dados podem ser selecionadas a partir de suas características, como organismo, nome e identificador da sequência, se foram processadas ou não, se tem conclusões atribuídas, tipo de evidência, *pipeline* executado e por programa executado. A plataforma permite inclusive a busca de sequências por resultados de processamentos anteriores. Ao escolher mais de um parâmetro, é preciso definir a operação que deve ser aplicada entre eles (união, intersecção ou diferença). As sequências selecionadas são exibidas em uma tabela na mesma tela. A figura 3.13 ilustra a tela inicial de busca e o resultado após a busca inicial. Uma

vez realizada uma busca podem, ser executadas várias operações: criar um *pipeline* de processamento, processar as sequências, visualizar individualmente uma, analisar graficamente, excluir sequências da seleção ou limpar a seleção, que descreveremos a seguir.

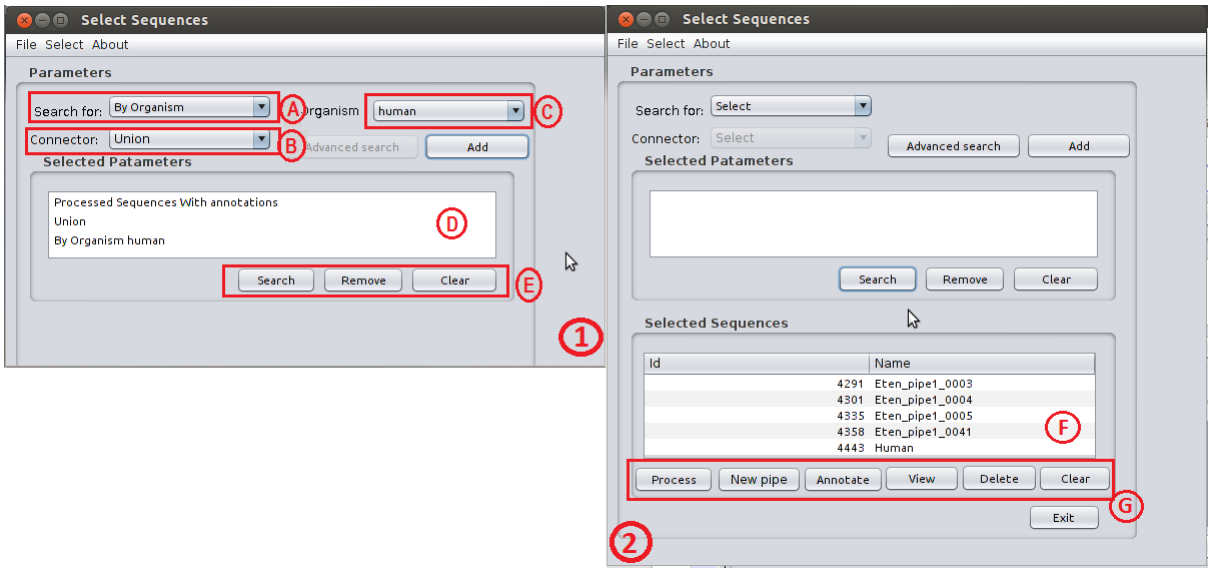


Figura 3.13: Tela de seleção do sistema (1) e tela com sequências selecionadas (2). A partir desta o usuário seleciona as sequências para análise. Os parâmetros disponíveis para análise são exibidos no menu (A). Caso o parâmetro selecionado possua opções, estas serão exibidas no menu ao lado(C). Uma vez selecionados, os parâmetro são adicionados pela tecla *Add*. No exemplo, a seleção do parâmetro organismo exibe os organismos armazenados na base de dados. Caso o usuário queira pesquisar por mais de um parâmetro, entre as opções escolhidas é necessário selecionar um conector (B). A consulta que está sendo formada pelo usuário é exibida de em forma de texto (D). As ações possíveis para a consulta formada são pesquisar, remover um item da pesquisa ou apagar a pesquisa inteira (E). O resultado da seleção é exibido na mesma tela de seleção (F) e o usuário pode executar um conjunto de ações com as sequências (G).

Novas buscas podem ser feitas após uma seleção inicial. A partir da mesma janela, o usuário pode realizar novas buscas, restringindo-as ou não às sequências previamente selecionadas. A figura 3.14 ilustra o processo de nova seleção quando se tem sequências previamente selecionadas.

3.2.3 Visão geral de uma sequência

A partir da seleção do botão *View* será lançada a janela de visualização da sequência (figura 3.15), indicando o nome da sequência, seu tipo (nucleotidica ou proteica) e seu organismo. Quatro botões (*Overview*, *Evidences*, *Annotations*, *Close*) permitem fechar essa janela ou am-

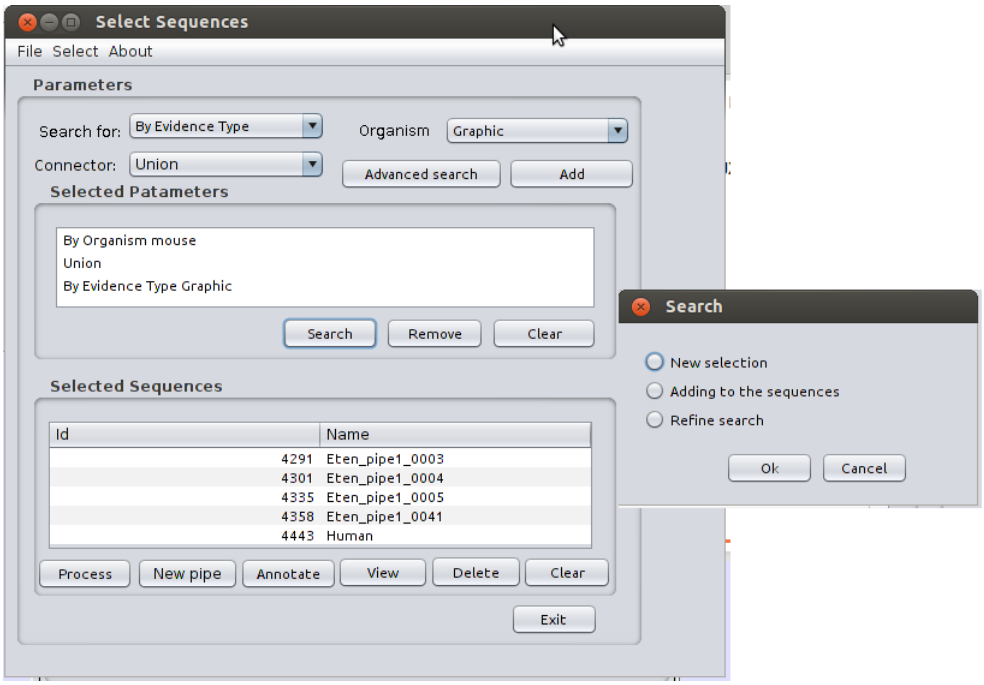


Figura 3.14: Nova seleção com sequências previamente selecionadas. O sistema pergunta o que o usuário deseja fazer: refinar a pesquisa, adicionar as sequências selecionadas ao resultado anterior ou realizar nova consulta,

pliar a visão para a visualização de diferentes conjuntos de dados, como: (B) número de bases e porcentagens (*Overview*), (C) os dados resultantes do processamento da sequência (*Evidences*); (D) as conclusões sobre a funcionalidade, localização da sequência (*Annotations*). A figura 3.16 ilustra a tela com todas as características da sequência sendo exibidas.

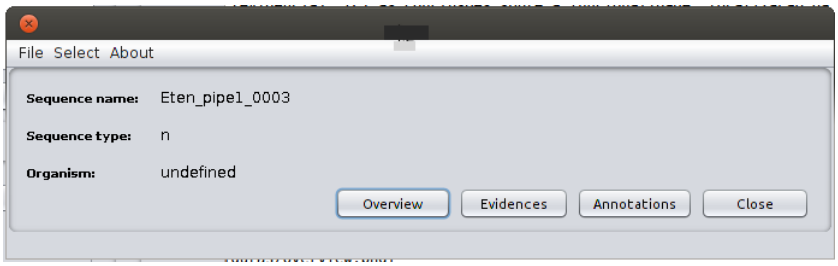


Figura 3.15: Tela que exibe a visão geral de uma sequência selecionada.

3.2.4 Processamento das sequências

O botão *New pipe* permite ao usuário criar novos pipelines a partir da ferramenta gráfica do *EGene*, o *CoEd*. Após a criação, o novo pipeline é inserido na base de dados. O botão *Process*

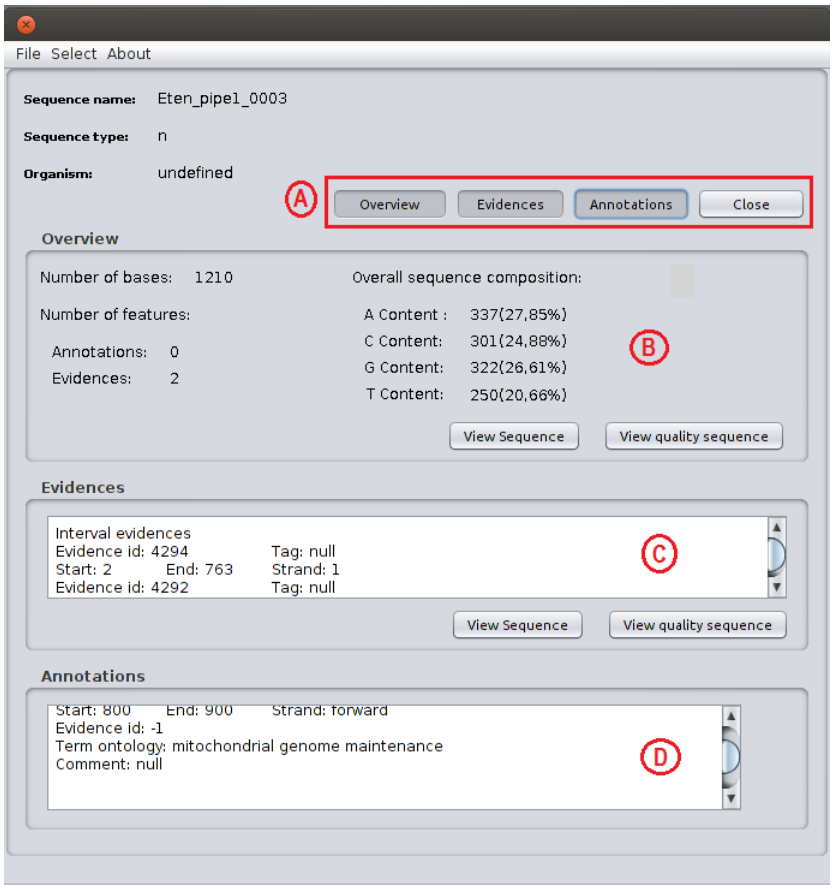


Figura 3.16: Tela que exibe a visão geral de uma sequência selecionada. A tela disponibiliza um conjunto de botões (A) para a visualização das cacterísticas da sequência, que exibem ou escondem os painéis que mostram um tipo de característica da sequência, que são seus dados gerais (B), suas evidências (C) e conclusões (D). Na figura, todas as informações da sequência são exibidas.

determina a realização de um pipeline para inclusão de novas informações na sequência. A partir deste o usuário seleciona o pipeline de processamento e o executa. O processamento é feito através da plataforma *EGene*.

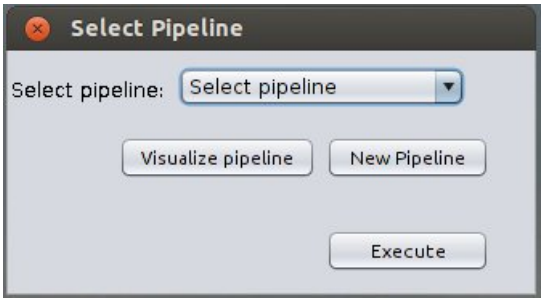


Figura 3.17: O sistema lista os *pipelines* armazenados no banco de dados. O usuário pode selecionar o *pipeline* de interesse e visualizá-lo por meio do *CoEd*. O usuário pode também construir um novo, caso os *pipelines* existentes não atendam as suas necessidades.

3.2.5 Análise individual

Ao analisar individualmente uma sequência, suas evidências (resultados de programas), conclusões e programas utilizados para processamento são exibidas graficamente para o usuário. A figura 3.18 ilustra as características de uma sequência. Cada tipo de característica é exibida em um painel distinto.

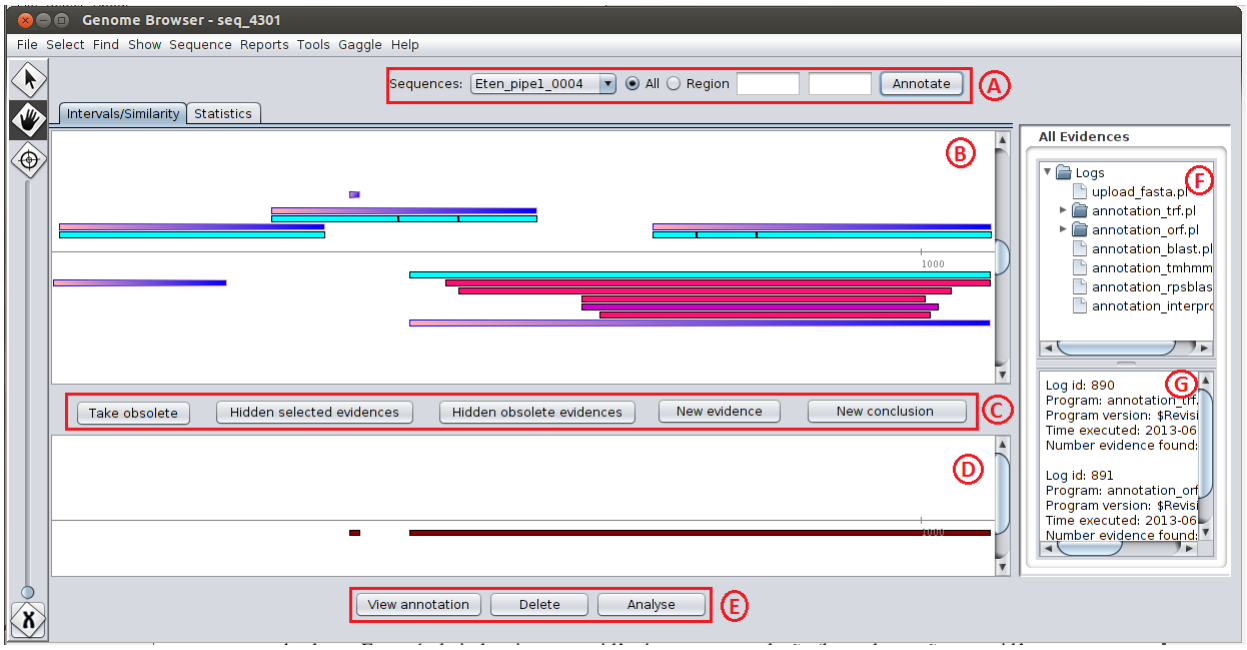


Figura 3.18: Tela de visualização de sequência. O usuário tem a opção de visualizar a sequência inteira ou apenas uma região específica (A). Selecionada a sequência, são exibidas suas evidências (B), conclusões atribuídas (D) e componentes que realizaram a análise (F). Em relação às evidências, são disponibilizadas ações como torná-la obsoleta, invisível e esconder evidências obsoletas. É possível ainda criar uma evidência ou uma conclusão (baseada ou não em evidências) (C). Para as conclusões, são disponibilizadas as ações de visualização, deleção ou análise (F). A análise de uma conclusão se relaciona com o processo de curagem. As análises realizadas na sequência são exibidas através de uma árvore (F) em ordem de execução. Ao selecionar uma, são exibidas sua descrição e a quantidade de evidências encontradas por ela (G).

Para a sequência selecionada, são disponibilizadas ações como exportar a sequência em arquivo fasta ou em arquivos de anotação (*Feature Table, GFF* ou *GenBank*), visualizar a composição nucleotídica da sequência, ter uma visão geral das características sequência (quantidade de evidências encontradas, de conclusões atribuídas, entre outras), visualizar evidências selecionadas e visualizar e/ou exportar as bases das evidências selecionadas em formato fasta.

Seleção de um evidência

Ao selecionar uma evidência, são disponibilizadas ações como visualizar suas características (coordenadas na sequência, análise que a encontrou, resultados da análise e subevidências, caso existam), sua composição nucleotídica e sequência de aminoácidos (caso a região seja codificadora de proteína). A figura 3.19 ilustra as propriedades de uma evidência selecionada.

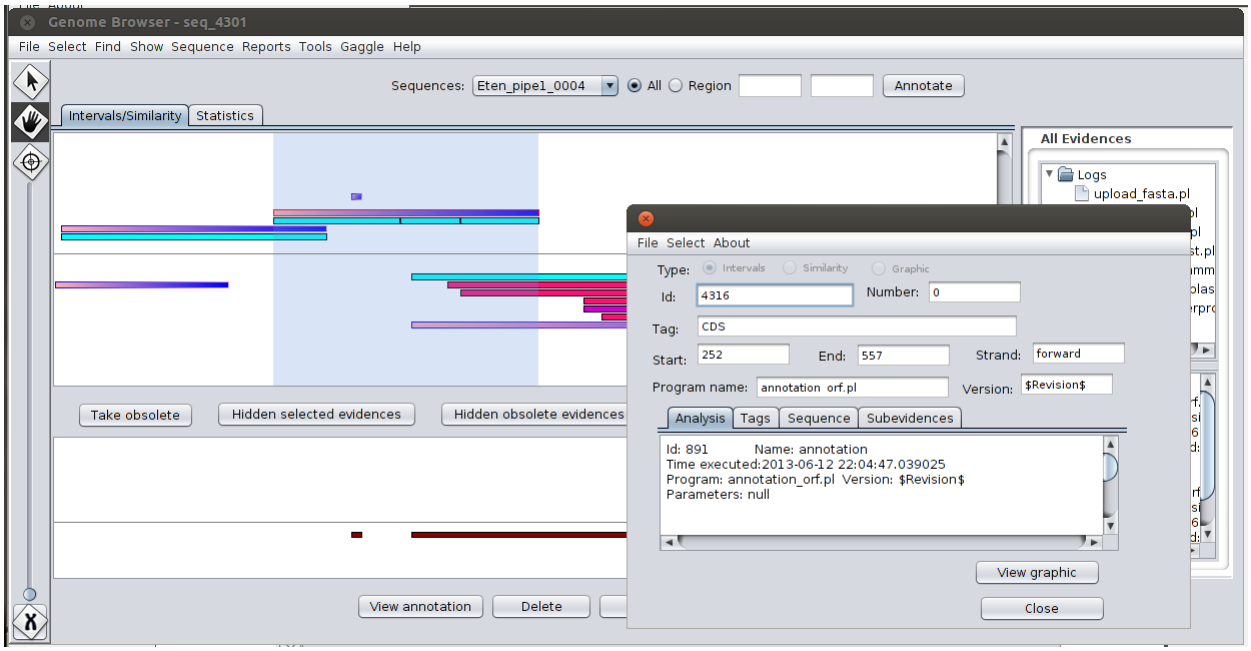


Figura 3.19: Visualização das propriedades de uma evidência. A parte superior do painel ilustra as coordenadas da evidência na sequência, seu tipo e o programa que a encontrou. O painel mostra suas demais características. A primeira aba descreve a análise que encontrou a evidência, a segunda ilustra das propriedades atribuídas a ela pelo programa, a terceira ilustra a sequência nucleotídica da região (permite que seja exportada) e a última mostra as subevidências (evidências encontradas a partir dessa evidência) caso existam.

O sistema permite ao usuário realizar análises em evidências selecionadas. Para tanto, são disponibilizadas duas opções: através de ferramentas online (para análises rápidas, em que o usuário deseja apenas visualizar o resultado) ou através do *EGene*. No processamento online, o sistema abre um navegador e redireciona o mesmo para a página da ferramenta. Nesta opção, apenas uma evidência é analisada por vez. Caso o *EGene* seja escolhido como ferramenta de processamento, o processo acontece da mesma forma descrita anteriormente. A figura 3.20 ilustra as operações possíveis de serem realizadas em uma evidência selecionada. Na figura, a opção de processamento foi selecionada, mostrando que o usuário pode processar uma evi-

dência a partir de um *pipeline* ou pela execução de um único programa, feita através de uma ferramenta online ou utilizando o *EGene*.

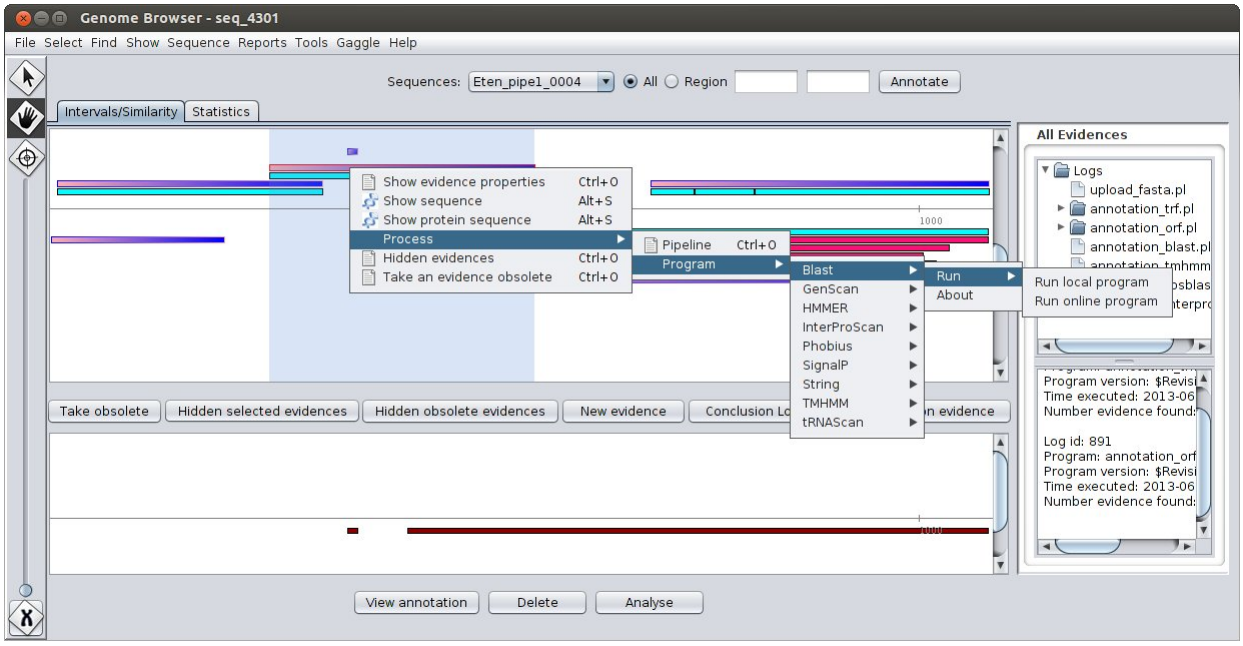


Figura 3.20: Ao selecionar uma evidência, o usuário pode visualizar suas propriedades, sua sequência nucloetídica e proteíca (caso seja uma região codificadora de proteína), processá-la, torná-la invisível ou obsoleta. A figura ilustra a seleção da opção de processamento.

O sistema possibilita ainda que o usuário torne uma evidência obsoleta, ou seja, permite ao usuário indicar que uma evidência não é mais do seu interesse. Essa opção é disponibilizada devido ao fato de não ser possível apagar uma evidência, pois são resultados da execução de um programa e podem ser utilizados no processo de curagem. Essa opção torna obsoleta a evidência e suas subevidências. O usuário pode também esconder evidências. A figura 3.21 ilustra uma evidência obsoleta.

Atribuição de conclusões

As conclusões são associadas a regiões da sequência. Para definir a região o usuário deve indicar um conjunto de evidências cuja a localização determinará a região. A figura 3.22 ilustra o processo de atribuição de de conclusão.

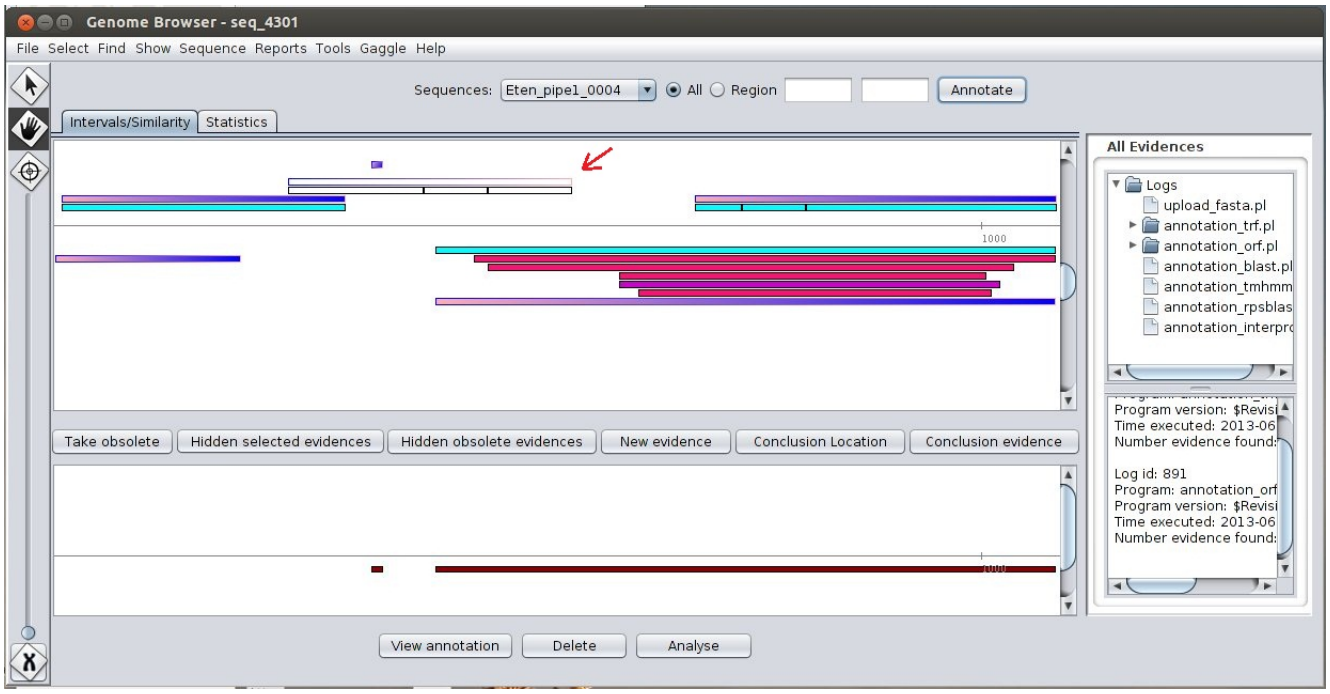


Figura 3.21: Tela de visualização com evidência obsoleta. A ação de tornar uma evidência obsoleta faz com que a mesma seja exibida de forma diferente das demais, pois somente suas bordas ficam visíveis. Essa opção torna obsoleta a evidencia e suas subevidências.

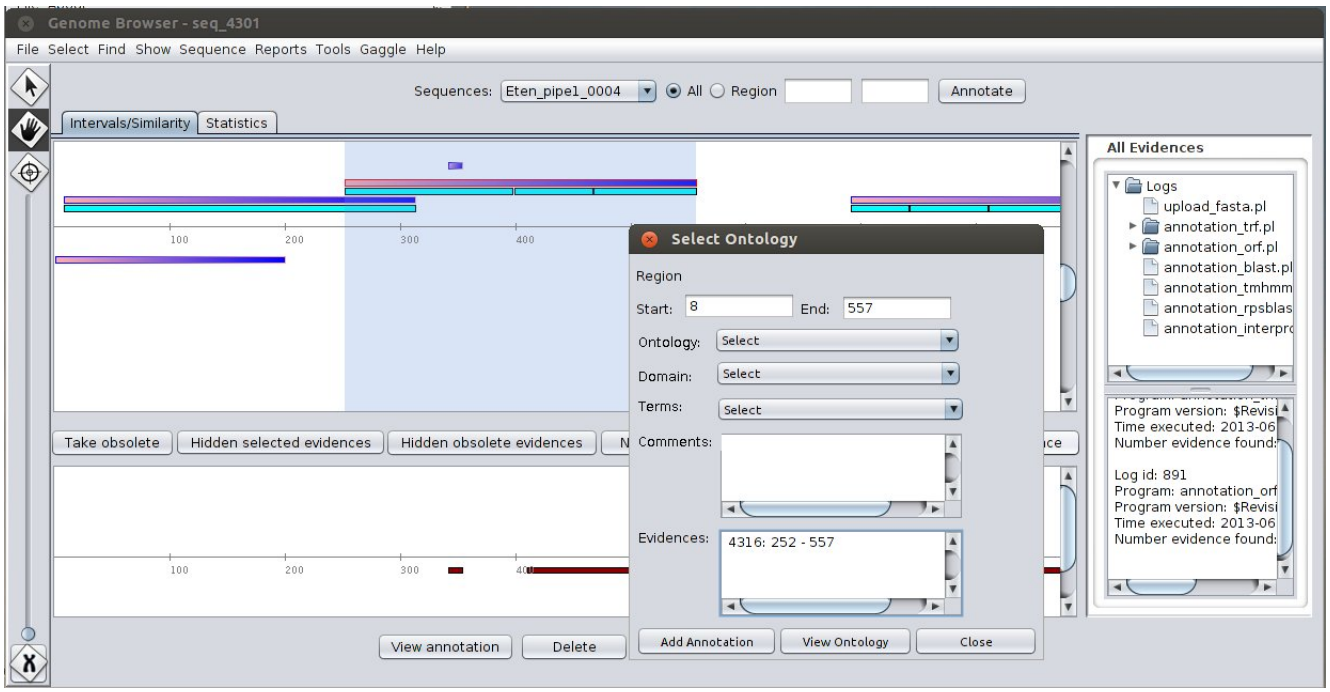


Figura 3.22: Atribuição de conclusão. O usuário seleciona a(s) evidência(s) e a opção de adicionar conclusão. O sistema calcula a região que engloba todas as evidências e a define como região de conclusão. O usuário então deve selecionar o termo de ontologia e adicionar algum comentário, caso ache necessário.

Seleção de conclusão e realização de curagem

Ao selecionar uma conclusão, o usuário pode visualizar suas propriedades (termo de ontologia atribuído, comentários adicionados, quem gerou), realizar edição, deleção ou curagem. O processo de curagem consiste na confirmação ou não das informações atribuídas. Após esse processo, as conclusões analisadas mudam de cor, com o objetivo de distinguí-la das demais. A figura 3.23 ilustra a visualização de uma conclusão.

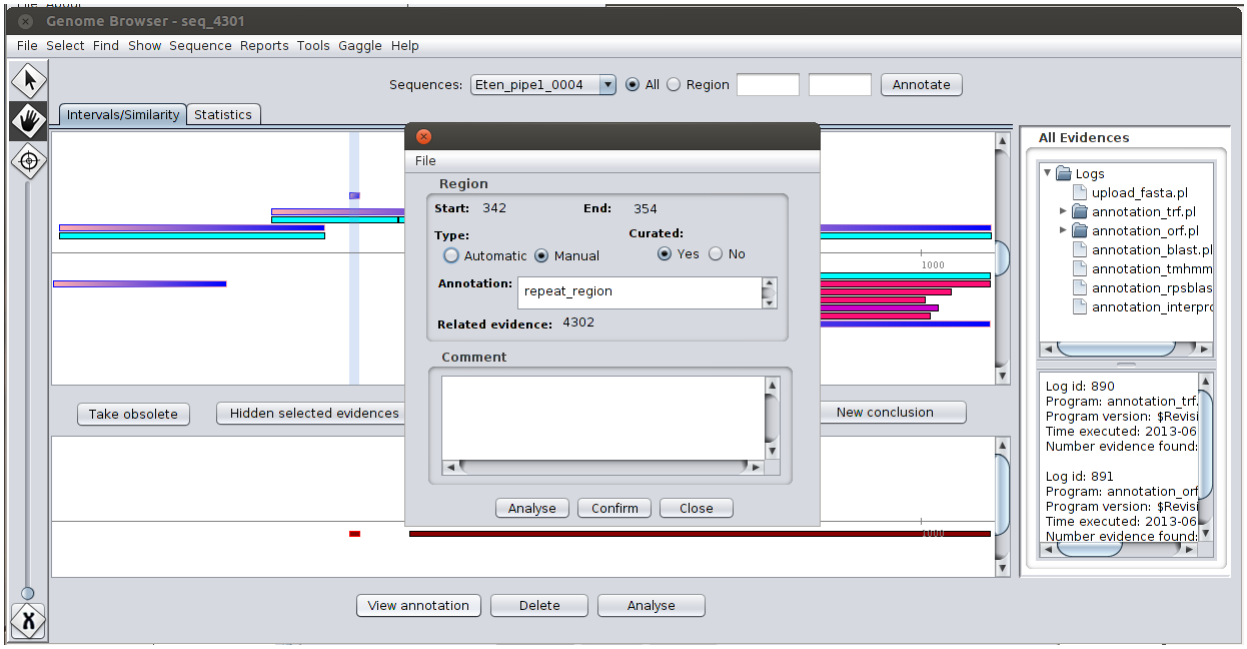


Figura 3.23: Visualização das características de uma conclusão. É permitido ao usuário analisar essa conclusão (permite a edição das características) ou confirmá-la. Ao confirmar uma conclusão, a mesma é considerada curada, o que faz que a sua cor no painel mude, de forma a distinguí-la das não curadas.

3.2.6 Validação da plataforma

A validação da plataforma foi realizada juntamente com o professor Dr. Arthur Gruber, a partir da execução de *pipelines* com dados reais (*contigs* da bactéria *Photorabdus luminescens*) e análise de resultados. Para uma validação inicial, foram executados dois *pipelines* com o mesmo conjunto de entrada, um construído com a versão original do *EGene* (que recebe os dados em arquivo e devolve os resultados igualmente em arquivo) e outro com a versão integrada com o banco de dados. O objetivo dessa validação foi verificar se o mapeamento dos dados no

Chado e os métodos da camada de representação foram implementadas corretamente. Para ambos *pipelines* foram gerados arquivos em formatos *Feature Table* a partir do resultado dos processamentos para fins de comparação. Observamos que as saídas eram iguais.

Uma segunda validação foi realizada por meio execução de consultas *Sql* construídas a partir de informações fornecidas pelo professor Arthur, também com o objetivo de validação do mapeamento e para verificação a utilidade da integração do *EGene* a uma base de dados. Foram elaboradas consultas que pesquisavam por evidências com determinadas características, como uma região de repetição com determinado período ou genes específicos.

Para a validação da interface visual, as sequências da bactéria foram selecionadas da base de dados e exibidas na tela de seleção. Todas as ações, com a exceção do processamento a partir do *CoEd* foram testadas e funcionam sem erros aparentes.

Capítulo 4

Conclusão

Nesse trabalho apresentamos a plataforma Pato, que permite ao usuário desempenhar as tarefas necessárias para a anotação de sequências genômicas a partir de uma interface gráfica. Através da plataforma o usuário consegue selecionar sequências por meio de buscas, processá-las, analisá-las graficamente e adicionar conclusões a respeito das regiões genômicas em um mesmo ambiente. Nossa pesquisa na literatura não revelou nenhuma ferramenta que permita ao usuário executar todo o processo de anotação em um único lugar, em geral as sequências são processadas e os resultados são retornados em formatos de arquivos, os quais podem ser abertos em ferramentas gráficas para a análise e anotação. A ferramenta que mais se aproxima é o *GenDB*, que permite ao usuário processar e analisar suas sequências, mas não permite seleção de sequências e possui um modelo de dados simples, com apenas tres elementos.

A plataforma Pato está conectada a um banco de dados, o que garante grande flexibilidade na busca e seleção de sequências e potencializa o uso dos dados gerados por outras plataformas. A flexibilidade é um dos diferenciais da plataforma. Para a busca de sequências, são disponibilizados os seguintes parâmetros: buscar todas as sequências, somente as sequências brutas (sequências sem processamento), sequências processadas com conclusões atribuídas, sequências processadas sem conclusões atribuídas, pelo tipo de evidência, organismo da sequência, *pipeline* executado, identificador ou nome da sequência ou resultado de algum programa executado. Os campos de busca são disponibilizados a partir dos resultados do programa selecionado pelo usuário. Por exemplo, se o usuário seleciona o programa blast, os campos disponíveis para busca são *evalue* e similaridade.

A validação inicial da plataforma foi feita a partir de dados reais da bactéria *Photorabdus luminescens*. Os testes mostraram que as estensões realizadas no *EGene*, no *Chado* e no visualizador foram robustas e a plataforma não demonstrou defeitos aparentes.

Como trabalho futuro podemos citar a habilitação para buscas de evidencias na interface visual. Para a efetivo uso para a anotação de genomas, é preciso implementar na interface a busca de características de um genoma, uma vez que atualmente só é possível executar buscas por sequências. Para um dado genoma, não é possível fazer um processamento de evidências específicas da mesma maneira que para sequências. A modelagem do *Chado* e do *EGene* permitem a anotação de genomas, porém a interface apenas permite a seleção e visualização de sequências, e não de parte de genomas.

A interface não permite que o usuário realize buscas de características por meio do visualizador. Para o funcionamento completo da capacidade de busca, seria interessante permitir que o usuário realizasse buscas no visualizador, e decidisse o que fazer os resultados da consulta. Por exemplo, o usuário gostaria de selecionar ou deixar visível somente as características de um determinado tipo, como as evidências que foram utilizadas para a atribuição de conclusão ou que foram encontradas por um determinado programa.

Acreditamos que esta plataforma será uma ferramenta valiosa para projetos de sequenciamento e anotação futuros.

Apêndice A

Mapeamento de dados

Os módulos do esquema *Chado* utilizados da plataforma são: *Sequence* (módulo central do esquema), *Companalysis* (armazenamento das análises computacionais realizadas) e *Controlled Vocabulary* (armazena as ontologias e vocabulários controlados), como descrito no capítulo 3. Por se relacionarem diretamente com os módulos escolhidos, os módulos *Organism* e *General* foram selecionados para a plataforma. Para a completa adequação do *Chado* a modelagem da plataforma, foram criados três módulos: *Sequencing project*, *Pipeline* e *Pre-processing*. Aqui detalharemos um pouco mais cada módulo.

O módulo *Sequence* como o próprio nome sugere, trata da sequência. Como descrito, o *Chado* considera toda sequência (ou parte dela) como uma *feature*. O módulo *Sequence* é capaz de representar *features*, o relacionamento entre elas, a localização de uma em relação a sua referência, propriedades e termos de ontologias atribuídos. A figura A.1 ilustra as tabelas do módulo. A tabela central do módulo é a *feature*. Cada *feature* é tipificada por um termo de ontologia (relacionamento este módulo com o de vocabulário controlado). Os relacionamentos entre features são armazenados na tabela *feature_relationship* e definidos igualmente por termos de ontologia. A localização das *features* é feita a partir da tabela *featureloc*, onde são definidas as coordenadas da localização. As propriedades das *features* são definidas a partir de termos de ontologia e armazenadas na tabela *featureprop*. Por fim, os termos de ontologia atribuídos às conclusões são armazenados em *feature_cvterm*. Essa tabela foi alterada de forma a armazenar as demais informações de uma conclusão. Os campos adicionados a essa tabela foram *type*, *curator* e *curated*, que se referem respectivamentes ao tipo da conclusão (manual

ou automática), usuário que gerou a anotação (se for automática, esse campo é preenchido com o nome do programa) e se a conclusão é curada (se foi analisada por um especialista). Esses campos são obrigatórios. Os comentários da conclusão são adicionados como propriedade da conclusão.

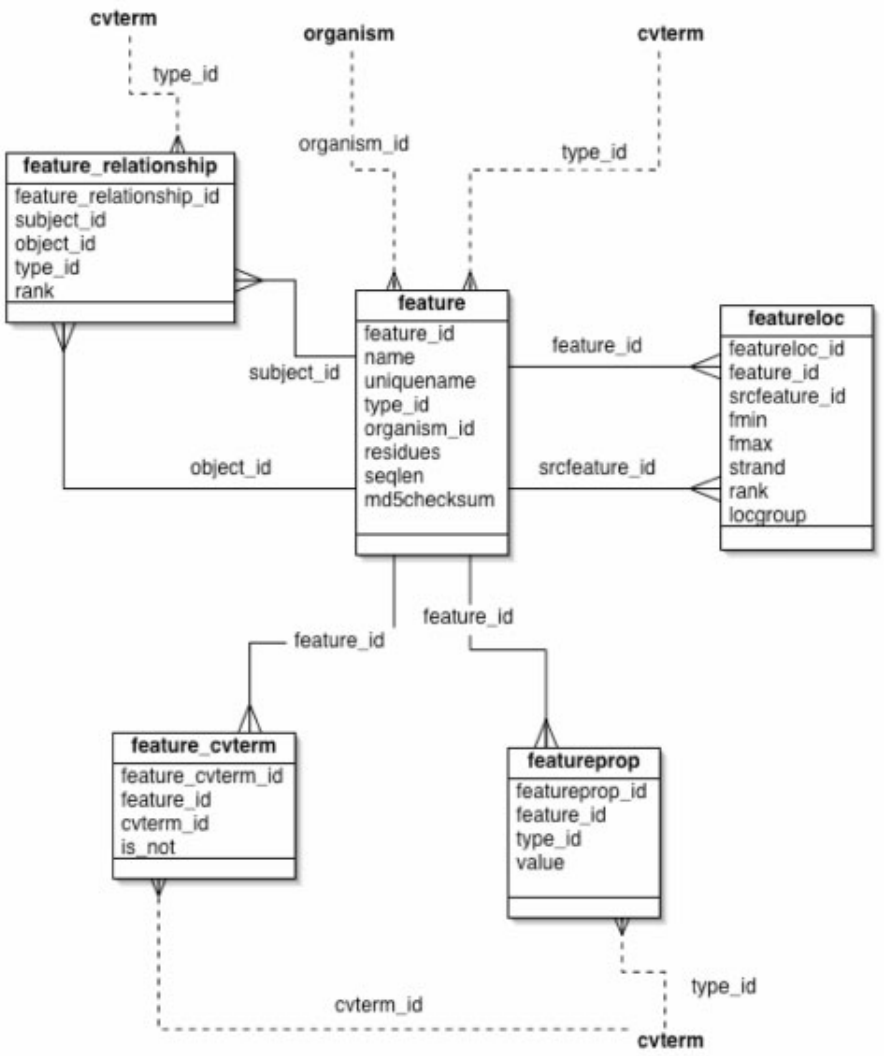


Figura A.1: Esquema do módulo *Sequence* do esquema *Chado*.

O módulo *Organism* armazena as informações a respeito do organismo, como mostra a figura A.2, bem como suas propriedades e representações em outros bancos. Como os projetos de sequenciamento podem ter organismos incialmente não definidos (por exemplo, os projetos de metagenômica sequenciam o material genético de uma ambiente. Inicialmente não se sabe quais organismos estão presentes na amostra), foi inserido um organismo no banco para representar essa situação.

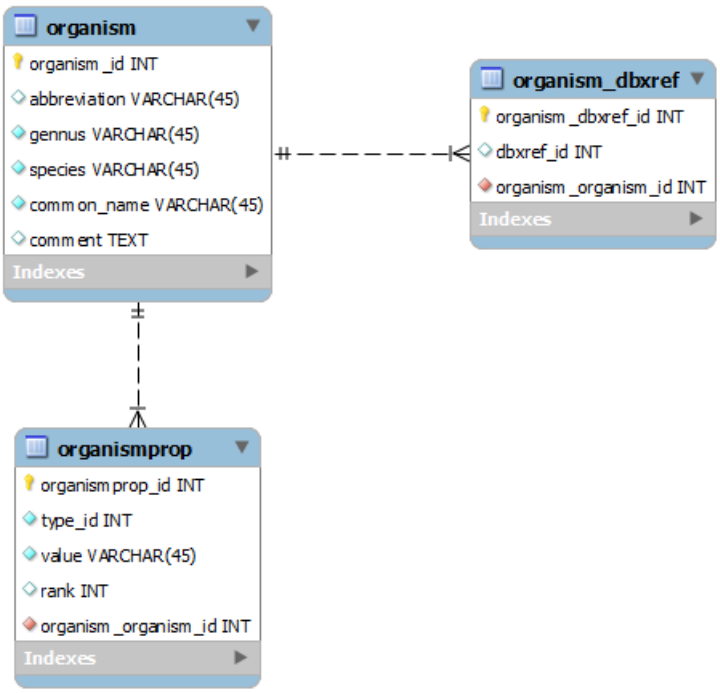


Figura A.2: Esquema do módulo *Organism*.

O módulo *Companalysis* trata das análises realizadas na sequência. A figura A.3 ilustra as tabelas do esquema. Como descrito, quando um *log* é registrado, as informações nome e versão do componente, data do processamento, nome e versão dos programas utilizados são armazenadas. A tabela *analysis* armazena o componente utilizado. Nela são armazenados os nome, versão e data do processamento. Os campos do programa se relacionam com o programa auxiliar utilizado pelo componente. Esse programa é igualmente armazenado como uma análise. Originalmente, o módulo *Companalysis* não possui recursos para definir relacionamentos entre análises. Para tornar esse tipo de relacionamento possível, foi adicionada a tabela *analysis_relationship*. De forma similar à tabela *feature_relationship*, a tabela *analysis_relationship* relaciona duas análises a partir de um termo de ontologia.

O módulo *Controlled Vocabulary* armazena ontologias e vocabulários controlados e seus respectivos termos. A figura A.4 ilustra as tabelas desse módulo utilizadas pela plataforma. A tabela *cv* armazena as ontologias e vocabulários, sendo indentificados a partir de seu nome e descrição. A tabela *cvterm* armazena os termos dos vocabulários descritos na tabela *cv*.

O módulo *General* armazena identificações únicas para as entidades no banco. Essas identificações podem ser padronizações utilizadas no bancos de origem (por exemplo, o termo da

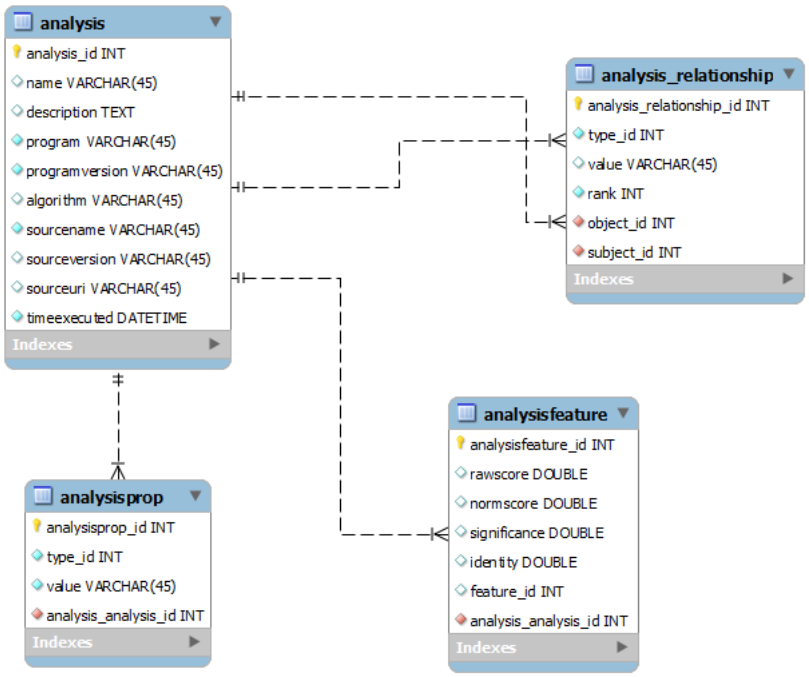


Figura A.3: Esquema do módulo *Companalysis* com a adição da tabela *analysis_relationship*.

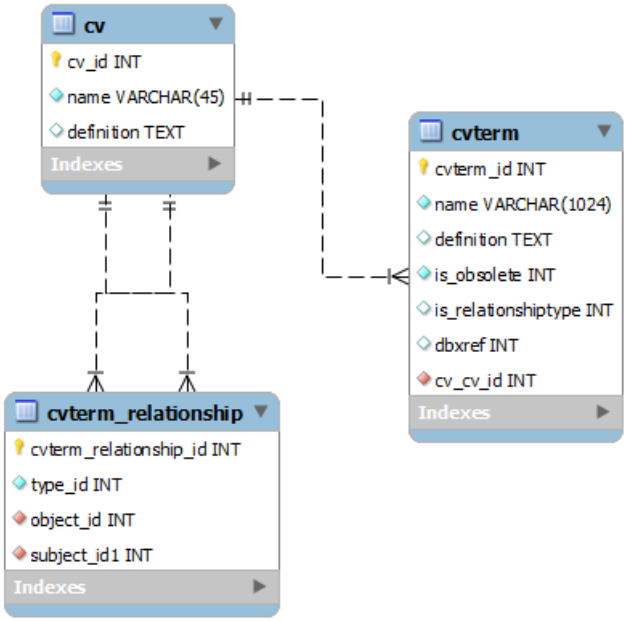
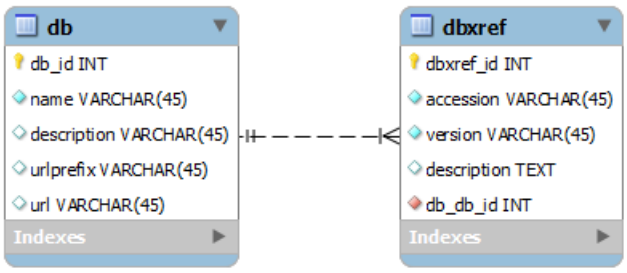


Figura A.4: Esquema do módulo *Controlled Vocabulary*.

ontologia de sequência gene tem o número de acesso SO:0000704. Esse identificador único é armazenado nas tabelas do módulo *General* e associado ao termo gene armazenado no módulo *cv*). A figura A ilustra as tabelas desse módulo.

O módulo de *Sequencing project* armazena dados dos projetos de sequenciamento dos quais



as sequências são oriundas. Esse módulo possui apenas uma tabela, que registra o nome do projeto, descrição, prefixo e data de criação (data de inserção das sequências no banco). A figura A.5 ilustra a tabela desse módulo. O projeto de sequenciamento ao qual a sequência pertence é definido a partir de uma propriedade (tabela *featureprop*) com o termo de ontologia com o termo *sequencing_project* (vocabulário controlado do *EGene*).

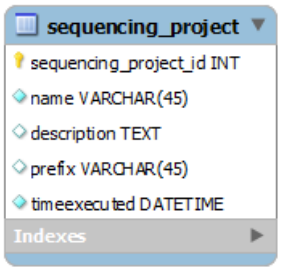


Figura A.5: Esquema do módulo de projeto de sequenciamento.

O módulo *Pipeline* armazena os *pipelines* criados pelo usuário e é capaz de relacioná-los com as sequências processadas por eles. A figura A.6 ilustra as tabelas desse módulo. Um *pipeline* pode ser executado por vários conjuntos de sequência, bem como um conjunto de sequência pode ser processado por vários *pipelines*.

O último módulo utilizado foi desenvolvido para o armazenamento do pré-processamento das sequências. Como dito, os componentes de pré-processamento são responsáveis pela execução de tarefas que preparam as sequências brutas (resultantes de um processo de sequenciamento) para a etapa de coleta de evidências. Abrange as tarefas de trimagem (corte das extremidades da sequência), mascaramento (troca das bases de uma região da sequência por *x*) e invalidação (invalidação de um subconjunto de sequências de acordo com um parâmetro de entrada, por exemplo, tamanho). Esse módulo associa um *log* a um operação (trimagem, mascaramento ou invalidação). Como cada operação armazena diferentes tipos de informação,

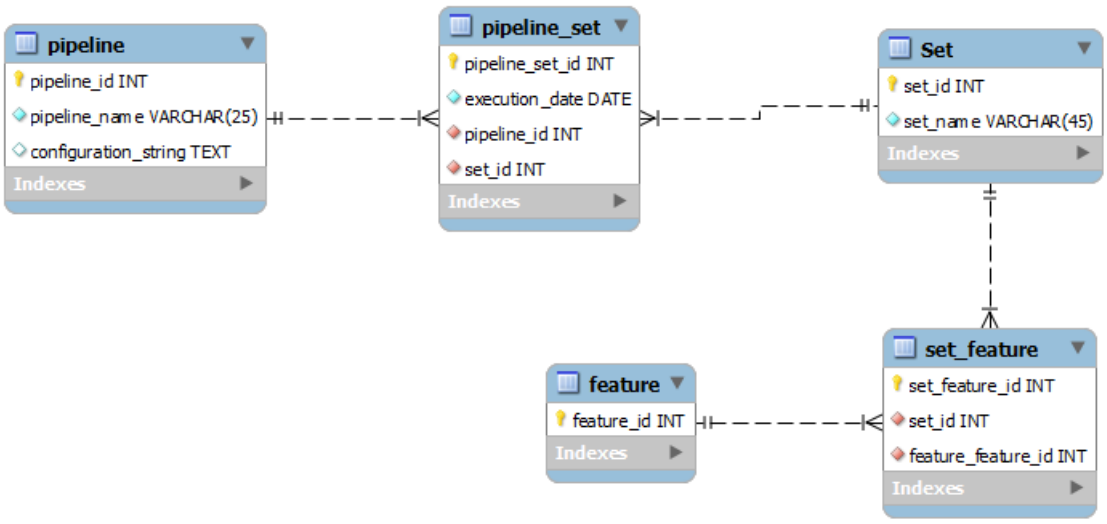


Figura A.6: Diagrama do módulo *pipeline*.

foram criadas três tabelas, uma tabela para cada tipo de operação. A figura A.7 exibe o módulo de pré-processamento.

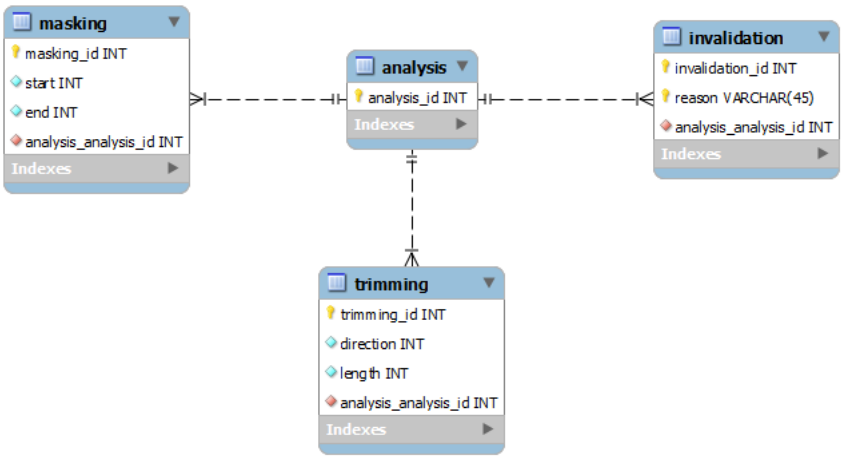


Figura A.7: Módulo criado para o armazenamento das operações de pré-processamento. Cada operação se relaciona com um *log* e, por isso, há o relacionamento de cada uma das tabelas com a tabela de análise.

Apêndice B

Termos do vocabulário controlado *egene_cv*

Como descrito no capítulo 3, devido a ausência de termos nos vocabulários controlados existentes para representar características específicas do *EGene*, foi desenvolvido o vocabulário controlado *egene_cv*. Este contém termos para descrever tipos de *features*, propriedades dos resultados gerados nos processamentos, propriedades e relacionamento entre as análises. Segue abaixo a lista dos termos do *egene_cv*.

Termos que descrevem o tipo de uma sequência:

a
n
intervals
graphic
statistic
similarity
conclusion_region

Termos que descrevem propriedades das sequências (a partir da execução de um componente):

type
number
protein_sequence
strand
minimum_size
orf_count
genetic_code
size
block_size
direction
version
predicted_TMHs
probability
probability_signalP
pep_sig
start_residue
end_residue
database

interpro_id
evidence_process
evidence_function
cleavage_site
query_frame
percent_matches
entropy
percent_indels
consensus_size
copy_number
period_size
classification
evidence_component
origin_file
origin_file_type
annotation
graphic_file
sequence_id
description
accession
database_code
subject_name
subject_id
alignment
anticodon_end
anticodon_start
codon_start
DB_id
DB_name
end
evalue
exon_number
exon_type
external_annotation
filtering
frame
interval
left
length
aminoacid
log_invalidation
masked_sequence
minimum_score
percent_id
primer_name
query_name
query_sequence
reason_for_invalidation

log_direction
right
seq_name
sequence
set_pipeline
pipeline_id
start
start_codon
subject_sequence
tagged_interval
trimmed_left
trimmed_right
trimming_direction
assembled_sequence
phd_singlet
other_vector
location_vector
singlet
cleavage_position1
cleavage_position2
SIGNAL
graphic_vector
cutoff
tRNAscan
intron_start
intron_end
intron_start_seq
intron_end_seq
HGT
threshold
note
target_identifier
target_name
target_type
target_class
target_description
target_start
target_end
truncated
gc_content
RNA_scan
bias
molecule_type
confidence
hairpin
tail
tail5
stem5

loop
stem3
tail3
description_interpro
RBS_pattern
new_start_codon
old_start_codon
old_position
position_shift
CDS_name
bitscore
hit_name
hit_description
TCDB_ID
TCDB_class
TCDB_subclass
TCDB_family
sequencing_project
locus_tag
window_size
step
data

Termos descrevem propriedades das análises:

program_arguments

Termos descrevem relacionamentos entre análises:

auxiliar_program

Referências Bibliográficas

[AJL+08] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, Peter Walter, John Wilson e Tim Hunt. *Molecular biology of the cell*. Garland Science, Taylor & Francis Group, fifth edição, 2008.

[BCMS07] Ezio Bartocci, Flavio Corradini, Emanuela Merelli e Lorenzo Scortichini. Bioowms: a web-based workflow management system for bioinformatics. 2007.

[BDE+12] Sarah W. Burge, Jennifer Daub, Ruth Eberhardt, John Tate, Lars Barquist, Eric P. Nawrocki, Sean R. Eddy, Paul P. Gardner e Alex Bateman. Rfam 11.0: 10 years of rna families. *Nucleic Acids Research*, 2012.

[Ben99] Gary Benson. Tandem repeats finder: a program to analyze dna sequences. *Nucleic Acids Research*, 1999.

[BK97] Chris Burge e Samuel Karlin. Prediction of complete gene structures in human genomic dna. *Journal Molecular Biology*, 1997.

[BKMC+11] Dennis A. Benson, Ilene Karsch-Mizrachi, Karen Clark, David J. Lipman, James Ostell e Eric W. Sayers. Genbank. *Nucleic Acids Research*, 2011.

[BKR+10] J Christopher Bare, Tie Koide, David J Reiss, Dan Tenenbaum e Nitin S. Baliga. Integration and visualization of systems biology data in context of the genome. *BMC Bioinformatics*, 2010.

[BO99] Jonathan H. Badger e Gary J. Olsen. Critica: Coding region identification tool invoking comparative analysis. *Protein coding prediction*, 1999.

[CCS01] Simon E. Cawley, Anthony I. Wirth C e Terence P. Speed. Phat - a gene finding program for plasmodium falciparum. *Molecular & Biochemical Parasitology*, 2001.

[CJKP05] Terry Clark, Josef Jurek, Gregory Kettler e Daphne Preuss. A structured interface to the object-oriented genomics unified schema for xml-formatted data. *Appl Bioinformatics*, 2005.

[Con00] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature American*, 2000.

[DHK+99] Arthur L. Delcher, Douglas Harmon, Simon Kasif, Owen White e Steven L. Salzberg. Improved microbial gene identification with glimmer. *Nucleic Acid Research*, 1999.

[DKM+05] Alan Mitchell Durham, André Yoshiaki Kashiwabara, Fernando T. G. Matsunaga, Paulo H. Ahagon, Flávia Rainone, Leonardo Varuzza e Arthur Gruber. Egene: a configurable pipeline generation system for automated sequence analysis. *Bioinformatics*, 2005.

[DSS+05] Gary H. Van Domselaar, Paul Stothard, Savita Shrivastava, Joseph A. Cruz, An-Chi Guo, Xiaoli Dong, Paul Lu, Duane Szafron, Russ Greiner e David S. Wishart. Basys: a web server for automated bacterialgenome annotation. *Nucleic Acids Research*, 2005.

[EHJV00] Gamma Erich, Richard Helm, Ralph Johnson e John Vlissides. Padrões de projeto. *Bookman*, 2000.

[ELM+05] Karen Eilbeck, Suzanna E Lewis, Christopher J Mungall, Mark Yandell, Lincoln Stein, Richard Durbin e Michael Ashburner. The sequence ontology: a tool for the unification of genome annotations. *Genome Biology*, 2005.

[emb05] Embl sequence format. http://www.bioperl.org/wiki/EMBL_sequence_format, 2005. [Online; accessed 16-Dec-2013].

[FAN+nd] Milene Ferro, Ricardo Yamamoto Abe, Rafael Moreira Neves, André Yoshiaki Kashiwabara, Luis Thibério L.D. Rangel, Alan Mitchell Durham e Arthur Gruber. Egene 2. n.d.

[FCE11] Robert D. Finn, Jody Clements e Sean R. Eddy. Hmmer web server: interactive sequence similarity searching. *Nucleic Acids Research*, 2011.

[fea12] The ddbj/embl/genbank feature table definition. http://www.insdc.org/files/feature_table.html, 2012. [Online; accessed 10-Aug-2013].

[FRG+07] Lance Feagan, Justin Rohrer, Alexander Garrett, Heather Amthauer, Ed Komp, David Johnson, Adam Hock, Terry Clark, Gerald Lushington, Gary Minden e Victor Frost. Bioinformatics process management: information flow via a computational journal. 2007.

[FvdBD+08] Mark WEJ Fiers, Ate van der Burgt, Erwin Datema, Joost CW de Groot e Roeland CHJ van Ham. High-throughput bioinformatics with the cyrille2 pipeline system. 2008.

[gmo08] Gmod. http://gmod.org/, 2008. [Online; accessed 10-Aug-2013].

[gtf13] Gff/gtf file format - definition and supported options. http://www.ensembl.org/info/website/upload/gff.html, 2013. [Online; accessed 13-Aug-2013].

[HCBS02] Alexander K. Hudek, Joseph Cheung, Andrew P. Boright e Stephen W. Scherer. Genescript: Dna sequence annotation pipeline. *Bioinformatics*, 2002.

[HJM+11] Sarah Hunter, Philip Jones, Alex Mitchell, Rolf Apweiler, Teresa K. Attwood, Alex Bateman, Thomas Bernard, David Binns, Peer Bork, Sarah Burge, Edouard de Castro, Penny Coggill, Matthew Corbett, Ujjwal Das, Louise Daugherty, Lauranne Duquenne, Robert D. Finn, Matthew Fraser, Julian Gough, Daniel Haft,

Nicolas Hulo, Daniel Kahn, Elizabeth Kelly, Ivica Letunic, David Lonsdale, Rodrigo Lopez, Martin Madera, John Maslen, Craig McAnulla, Jennifer McDowall, Conor McMenamin, Huaiyu Mi, Prudence Mutowo-Muellenet, Nicola Mulder, Darren Natale, Christine Orengo, Sebastien Pesseat, Marco Punta, Antony F. Quinn, Catherine Rivoire, Amaia Sangrador-Vegas, Jeremy D. Selengut, Christian J. A. Sigrist, Maxim Scheremetjew, John Tate, Manjulapramila Thimmajanarthanan, Paul D. Thomas, Cathy H. Wu, Corin Yeats e Siew-Yit Yong. Interpro in 2011: new developments in the family and domain prediction database. *Nucleic Acids Research*, 2011.

[HRmC+03] Shawn Hoon, Kiran Kumar Ratnapu, Jer ming Chia, Balamurugan Kumarasamy, Xiao Juguang, Michele Clamp, Arne Stabenau, Simon Potter, Laura Clarke e Elia Stupka. Biopipe: A flexible framework for protocol-based bioinformatics analysis. 2003.

[HSS+00] D. Hyatt, J. Snoddy, D. Schmoyer, G. Chen, K. Fischer, M. Parang, I. Vokler, S. Petrov, P. Locascio, V. Olman, Miriam Land e M. Shah. Improved analysis and annotation tools for whole-genome computational annotation and analysis: Grail-exp genome analysis toolkit and related analysis tools. *Genome Sequencing & Biology Meeting*, 2000.

[Iba03] Conrad V. Ibañez. Gus sb - a schema browser for the genomics unified schema(gus). Dissertação de Mestrado, 2003.

[Kas11] Andre Yoshiaki Kashiwabara. *MYOP/ToPS/SGEval: Um ambiente computacional para estudo sistemático de predição de genes*. Tese de Doutorado, 2011.

[KBK03] Roman Kolpakov, Ghizlane Bana e Gregory Kucherov. mreps: efficient and flexible detection of tandem repeats in dna. *Nucleic Acids Research*, 2003.

[KDC+11] Kamal Kumar, Valmik Desai, Li Cheng, Maxim Khitrov, Deepak Grover, Ravi Vijaya Satya, Chenggang Yu, Nela Zavaljevski e Jaques Reifman. Ages: A software system for microbial genome sequence annotation. *PLos one*, 2011.

[KGLB05] Liisa B Koski, Michael W Gray, B Franz Lang e Gertraud Burger. Autofact: An automatic functional annotation and classification tool. *BMC Bioinformatics*, 2005.

[KKS04] Lukas Kall, Anders Krogh e Erik L. L. Sonnhammer. A combined transmembrane topology and signal peptide prediction method. *Journal Molecular Biology*, 2004.

[Kor04] Ian Korf. Gene finding in novel genomes. *BMC Bioinformatics*, 2004.

[Kro97] A. Krogh. Two methods for improving performance of an hmm and their application for gene finding. *Molecular Biology*, 1997.

[Lap01] Hilmar Lapp. Biosql, 2001.

[LE97] Todd M. Lowe e Sean R. Eddy. trnascan-se: a program for improved detection of transfer rna genes in genomic sequence. *Nucleic Acids Research*, 1997.

[LHR+07] Karin Lagesen, Peter Hallin, Einar Andreas Rodland, Hans-Henrik Staerfeldt, Torbjorn Rognes e David W. Ussery. Rnammer: consistent and rapid annotation of ribosomal rna genes. *Nucleic Acids Research*, 2007.

[Lin01] Stein Lincoln. Genome annotation: from sequence to biology. *Nature Reviews Genetics*, 2001.

[LSH+02] SE Lewis, SMJ Searle, N Harris, M Gibson, V Iyer, J Richter, C Wiel, L Bayraktaroglu, E Birney, MA Crosby, JS Kaminker, BB Matthews, SE Prochnik, CD Smith, JL Tupy, GM Rubin, S Misra, CJ Mungall e ME Clamp. Apollo: a sequence annotation editor. *Genome Biology*, 2002.

[MEC07] Christopher J. Mungall, David B. Emmert e The FlyBase Consortium. A chado case study: an ontology-based modular schema for representing genome-associated biological information. *Bioinformatics*, 2007.

[MGM+03] Folker Meyer, Alexander Goesmann, Alice C. McHardy, Daniela Bartels, Thomas Bekel, Jorn Clausen, Jorn Kalinowski, Burkhard Linke, Oliver Rupp, Robert Giegerich e Alfred Puhler. Gendb - an open source genome annotation system for prokaryote genomes. *Nucleic Acid Research*, 2003.

[ML06] Ariane Machado-Lima. *Predição de RNAs não-codificantes e sua aplicação na busca do componente RNA da telomerase*. Tese de Doutorado, Novembro 2006.

[MLdPD08] Ariane Machado-Lima, Hernando A. del Portillo e Alan Mitchell Durham. Computational methods in noncoding rna research. *Mathematical Biology*, 2008.

[MPS04] W. H. Majoros, M. Pertea e S. L. Salzberg. Tigrscan and glimmerhmm: two open source ab initio eukaryotic gene-finders. *Bioinformatics*, 2004.

[NTI08] Hideki Noguchi, Takeaki Taniguchi e Takehiko Itoh. Metageneannotator: Detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. *DNA Research*, 2008.

[OAF+04] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R. Pocock, Anil Wipat e Peter Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. 2004.

[PBvHN11] Thomas Nordahl Petersen, Soren Brunak, Gunnar von Heijne e Henrik Nielsen. Signalp 4.0: discriminating signal peptides from transmembrane regions. *Nature Methods*, 2011.

[Pea06] Helen Pearson. What is a gene? 2006.

[PGY+12] Pjotr Prins, Naohisa Goto, Andrew Yates, Laurent Gautier, Scooter Willis, Christopher Fields e Toshiaki Katayama in. *Evolutionary Genomics: Statistical and Computational Methods*. Springer Science+Business Media, second edição, 2012.

[RPC+00] Kim Rutherford, Julian Parkhill, James Crook, Terry Horsnell, Peter Rice, Marie-Adele Rajandream e Bart Barrell. Artemis: sequence visualization and annotation. *Bioinformatics*, 2000.

[SDKW98] Steven L. Salzberg, Arthur L. Delcher, Simon Kasif e Owen White. Microbial gene identification using interpolated markov models. *Nucleic Acid Research*, 1998.

[Sea09] Mitchell E. Skinner e Andrew V. Uzilov Lincoln D. Stein et al. Jbrowse: A next-generation genome browser. *Genome Research*, 2009.

[SGM+90] Altschul SF, W Gish, W Miller, Myers EW e Lipman DJ. Basic local alignment search tool. *J. Mol. Biol.*, 1990.

[SHGnd] A.F.A. Smit, R. Hubley e P. Green. Repeatmasker. n.d.

[SHS+04] Sohrab P Shah, David YM He, Jessica N Sawkins, Jeffrey C Druce, Gerald Quon, Drew Lett, Grace XY Zheng, Tao Xu e BF Francis Ouellette. Pegasys: software for executing and integrating analyses of biological sequences. 2004.

[SKG+06] Mario Stanke, Oliver Keller, Irfan Gunduz, Alec Hayes, Stephan Waack e Burkhard Morgenstern. Augustus: ab initio prediction of alternative transcripts. *Nucleic Acids Research*, 2006.

[SMS+02] Lincoln D. Stein, Christopher Mungall, ShengQiang Shu, Michael Caudy, Marco Mangone, Allen Day, Elizabeth Nickerson, Jason E. Stajich, Todd W. Harris, Adrian Arva e Suzanna Lewis. The generic genome browser: A building block for a model organism system database. *Genome Research*, 2002.

[SOR09] Andrew C. Stewart, Brian Osborne e Timothy D. Read. Diya: a bacterial annotation pipeline for any genomics lab. *Bioinformatics*, 2009.

[Ste13] Lincoln Stein. Generic feature format version 3. http://www.sequenceontology.org/gff3.shtml, 2013. [Online; accessed 10-Aug-2013].

[SvHK98] Erik L.L. Sonnhammer, Gunnar von Heijne e Anders Krogh. A hidden markov model for predicting transmembrane helices in protein. *American Association for Artificial Intelligence*, 1998.

[TGM+10] David M. Tanenbaum, Johannes Goll, Sean Murphy, Prateek Kumar, Nikhat Zafar, Mathangi Thiagarajan, Ramana Madupu, Tanja Davidsen, Leonid Kagan, Saul Kravitz, Douglas B. Rusch e Shibu Yooseph. The jcvi standard operating procedure for annotating prokaryotic metagenomic shotgun sequencing data. *Standards in Genomic Sciences*, 2010.

[TS12] Todd J. Treangen e Steven L. Salzberg. Repetitive dna and next-generation sequencing: computational challenges and solutions. 2012.

[web13] Web of knowledge. http://apps.webofknowledge.com/, 2013. [Online; accessed 13-Aug-2013].

[YCZ+12] Jian Ye, George Coulouris, Irena Zaretskaya, Ioana Cutcutache, Steve Rozen e Thomas L Madden. Primer-blast: A tool to design target-specific primers for polymerase chain reaction. *BMC Bioinformatics*, 2012.

[YZD+08] Chenggang Yu, Nela Zavaljevski, Valmik Desai, Seth Johnson, Fred J Stevens e Jaques Reifman. The development of pipa: an integrated and automated pipeline for genome-wide protein function annotation. *BMC Bioinformatics*, 2008.