



Universidade Estadual de Santa Cruz

ZILTON JOSÉ MACIEL CORDEIRO JUNIOR

**APRIMORAMENTOS DA INTERFACE GRÁFICA PARA A INTEGRAÇÃO DE
SOLUÇÕES PARA RECONSTRUÇÃO DE ÁRVORES FILOGENÉTICAS – IGRAFU
2.0.**

ILHÉUS-BAHIA

2007

ZILTON JOSÉ MACIEL CORDEIRO JUNIOR

**APRIMORAMENTOS DA INTERFACE GRÁFICA PARA A INTEGRAÇÃO DE
SOLUÇÕES PARA RECONSTRUÇÃO DE ÁRVORES FILOGENÉTICAS – IGRAFU
2.0.**

Relatório apresentado, para obtenção da aprovação na
disciplina Estágio Supervisionado, à Universidade
Estadual de Santa Cruz.

Área de Concentração: Bioinformática

Orientadora: Prof. Martha Ximena Torres Delgado

ILHÉUS-BAHIA

2007

ZILTON JOSÉ MACIEL CORDEIRO JUNIOR

**APRIMORAMENTOS DA INTERFACE GRÁFICA PARA A INTEGRAÇÃO DE
SOLUÇÕES PARA RECONSTRUÇÃO DE ÁRVORES FILOGENÉTICAS – IGRAFU
2.0.**

Ilhéus-BA, 04/12/2007.

Martha Ximena Torres Delgado – Dsc.

UESC

(orientadora)

Carlos Pereira – Msc.

UESC

Dany Sanches – Dsc.

UESC

Professor da Disciplina Estágio Supervisionado

DEDICATÓRIA

Aos meus pais e irmãos, que sempre me apoiaram e incentivaram para que eu chegasse até esta etapa de minha vida, dedico.

AGRADECIMENTOS

Agradeço primeiramente a minha namorada e amigos pelos momentos de companhia durante esses anos de faculdade.

A professora Martha Ximena, por ter me dado a oportunidade de concluir este estágio e por ter sido, durante esses anos de convivência, uma ótima orientadora.

A Universidade Estadual de Santa Cruz, pela estrutura.

A todos que de alguma forma contribuíram para concretização deste trabalho.

APRIMORAMENTOS DA INTERFACE GRÁFICA PARA A INTEGRAÇÃO DE SOLUÇÕES PARA RECONSTRUÇÃO DE ÁRVORES FILOGENÉTICAS – IGRAFU 2.0.

RESUMO

Este projeto teve como objetivo o desenvolvimento de uma interface gráfica do tipo *front end* para os diferentes programas de reconstrução de árvores filogenéticas que estão sendo desenvolvidos no laboratório de bioinformática da UESC e para alguns dos mais importantes programas de domínio público, gerando assim uma ferramenta de fácil utilização para profissionais não especialistas em computação, a qual pode ser utilizada tanto em ambiente de cluster, como em ambiente desktop.

Palavras Chaves: Engenharia de software, Bioinformática, Reconstrução de Árvores Filogenéticas, Desenvolvimento de ferramentas gráficas em JAVA.

LISTA DE FIGURAS

Figura 1-	Representação gráfica de uma árvore filogenética	19
Figura 2-	Casos de uso da IGRAFU	27
Figura 3-	Diagrama de estados de navegação da IGRAFU	28
Figura 4-	Estrutura de pacotes do Eclipse	30
Figura 5-	Diagrama de classe do pacote br.uesc.computacao.estagio.aplicao.controlado	32
Figura 6-	Casos de uso da IGRAFU	40
Figura 7-	Tela de Splash	55
Figura 8-	Modo de execução	56
Figura 9-	Tela de escolha do bootstrap	57
Figura 10-	Tela do seqboot – Sequência	58
Figura 11-	Tela seqboot – Parâmetros	59
Figura 12-	Tela Consense	60
Figura 13-	Tela Metodos	61
Figura 14-	Tela Programas - MV	62
Figura 15-	Tela Programas – Distância	63
Figura 16-	Tela phyml – Sequência	64
Figura 17-	Tela phyml – Modelo – DNA	65
Figura 18-	Tela phyml – Modelo – AA	66
Figura 19-	Tela phyml – Árvore	67
Figura 20-	Tela phyml Tipo de Execução	68
Figura 21-	Tela DiGrafu Sequência	69
Figura 22-	Tela DiGrafu – Modelo – DNA	70
Figura 23-	Tela DiGrafu Local de Execução	71
Figura 24-	Tela Editor	72
Figura 25-	Tela Hypertree	73

LISTA DE TABELAS

Tabela 1 -	Cronograma de execução	15
Tabela 2 -	Casos de uso da lGrafu 2.0	39
Tabela 3 -	Interface de usuário e suas respectivas ações	54

LISTA DE SIGLAS

UESC – Universidade Estadual de Santa Cruz.

UML – *Unified Modeling Language*.

IGrafu – Interface Gráfica do Grupo de Reconstrução de Árvores Filogenéticas da UESC.

RUP - *Rational Unified Process*.

POO – Programação Orientada a Objeto.

IDE – *Integrated Development Environment*.

JUDE – *Java and UML Developers' Environment*.

API – *Application Programming Interface*.

MV – Máxima Verossimilhança.

MVC – *Model View Controller*.

SUMÁRIO

	RESUMO	V
	LISTA DE FIGURAS.....	VI
	LISTA DE TABELAS.....	VII
	LISTA DE SIGLAS.....	VIII
1	INTRODUÇÃO	10
1.1	Área de Atuação.....	11
1.2	Identificação do Problema.....	11
1.3	Objetivos.....	12
1.4	Metodologia.....	13
1.5	Cronograma.....	15
1.6	Justificativa e Contribuições.....	15
1.7	Organização Deste Relatório.....	16
2	FUNDAMENTOS TEÓRICOS	17
2.1	Bioinformática.....	17
2.2	Métodos Filogenéticos.....	18
2.2.1	Máxima Verossimilhança.....	18
2.2.2	Distância.....	18
2.3	Árvore Filogenética.....	19
2.4	<i>Bootstrap</i>	20
3	FERRAMENTAS E MÉTODOS	21
3.1	<i>Rational Unified Process</i>	21
3.2	Programação Orientada a Objeto.....	21
3.3	Java.....	22
3.4	UML.....	23
3.5	JUDE.....	24
3.6	Eclipse.....	24
4	COMPOSIÇÃO DO SOFTWARE	25
4.1	Seqboot.....	25
4.2	Consense.....	25
4.3	DiGrafu.....	25
4.4	Phyml.....	25
4.5	Hypertree.....	25
5	DESENVOLVIMENTO	26
5.1	Elaboração.....	26
5.2	Implementação do Projeto.....	29
6	TESTES	33
7	RESULTADOS E CONCLUSÕES	34
8	TRABALHOS FUTUROS	35
	REFERÊNCIAS	36
	APÊNDICE 1 – Especificação de Requisitos de Software	37
	APÊNDICE 2 – Telas do Sistema e Controles de Navegação	51

1 INTRODUÇÃO

A diversidade biológica, ou seja, as diferenças entre os grupos de organismos é entendida hoje como o resultado do processo de evolução. Os seres vivos não são entidades estáticas, mas se transformam ao longo de gerações sob influência do meio. Uma árvore filogenética é uma representação gráfica em forma de árvore, com vértices e arestas, em que todos os indivíduos nela presentes partem de um ancestral comum, a raiz. Cada aresta da árvore representa uma mutação na espécie, cada vértice, um ponto na escala evolutiva, e cada folha uma espécie diferente. A proximidade entre as folhas indica que aquelas espécies têm uma certa proximidade evolutiva.

O objetivo da análise filogenética é inferir a relação evolucionária correta entre três ou mais espécies contemporâneas de dados de seqüência (pode ser de aminoácidos, nucleotídeos, códon, etc.) de membros representativos de cada espécie (SCHADT et al., 1998). As árvores filogenéticas são importantes para compreender a história evolutiva das espécies, ajudar no desenvolvimento de vacinas, predição de novos genes, estudo da biodiversidade, entendimento da biologia macrobiana, usos biotecnológicos como na ajuda de produção de novas drogas para propósitos agrícolas e médicos, etc.

O desenvolvimento de uma ferramenta de reconstrução de árvores filogenéticas para a Universidade Estadual de Santa Cruz é de extrema importância

para as pesquisas nas áreas de genética molecular e bioinformática, já que, à medida que essas pesquisas avançam, principalmente no estudo da biodiversidade da mata atlântica, novas espécies são descobertas. Por isso é fundamental realizar o estudo das relações dessas espécies com outras já existentes. De maneira geral, utilizar uma ferramenta computacional com uma interface gráfica amigável facilita muito o trabalho final do usuário, visto que o tempo de trabalho será reduzido, os resultados finais serão melhor visualizados e o entendimento geral do problema poderá ser melhor estudado. Neste caso, particularmente, onde os principais usuários desta ferramenta serão profissionais não especialistas na área de computação é de extrema necessidade o desenvolvimento de uma interface gráfica fácil de manipular e totalmente auto-explicativa.

1.1 Área de Atuação

A área de atuação é a bioinformática, focado em reconstrução de árvores filogenéticas.

1.2 Identificação do problema

Para a reconstrução de árvores filogenéticas existem vários métodos como Parcimônia, Distância e Máxima Verossimilhança e estes métodos podem ser desenvolvidos utilizando diferentes técnicas como heurísticas próprias ou algoritmos estatísticos, como por exemplo, algoritmos genéticos (SCHADT et al., 1998; GUINDON et al. 2003; PRADO 2001; SCHMIDT, 2003). Atualmente, existem vários programas disponíveis para realizar reconstrução de árvores filogenéticas. Cada um

deles pode incluir mais de um método e mais de uma técnica. Como exemplos desses programas, temos: *Fastdnaml*, *Phyml*, *Ptp*, *Paml*, DiGrafu, etc. A maioria deles não tem uma interface gráfica ou quando a possui é uma interface rudimentar, que não dispõe de muitas facilidades para o usuário.

Ter uma interface gráfica, na qual envolva os principais programas utilizados, além de mecanismos de *bootstrap*, visualização e execução de alto desempenho, é muito importante para a evolução das pesquisas na área da bioinformática.

Além disso, um dos desafios nas pesquisas de reconstrução de árvores filogenéticas é desenvolver uma ferramenta que leve em consideração um número muito grande de seqüências; sendo assim, também é muito importante desenvolver algoritmos que permitam realizar uma visualização amigável para árvores muito grandes (SANDERSON et al., 2003).

1.3 Objetivos

a) Geral

Este projeto tem como objetivo fazer uma reengenharia de software além de continuar o desenvolvimento da IGrafu, a qual é uma interface gráfica para os diferentes programas de reconstrução de árvores filogenéticas que estão sendo desenvolvidos no laboratório de bioinformática da UESC e para alguns dos mais importantes *softwares* livres existentes, bem como incluir uma melhor visualização de árvores filogenéticas na tela, gerando assim uma ferramenta de fácil utilização para profissionais não especialistas em computação.

b) Específicos

- Disponibilizar o cálculo de *bootstrap* para todos os *softwares* que forem inclusos à IGrafu 2.0, através dos programas Seqboot e Consense que serão adicionados à IGrafu 2.0;
- Desenvolver um *software* destinado a ambientes de *Clusters* de alto desempenho e ambiente *desktop*, levando-se em conta que a IGrafu 2.0 é destinada à ambientes *Linux*;
- Desenvolver interface gráfica para os principais programas de reconstrução de árvores filogenéticas que utilizam os métodos:
 - Distância: Programa DiGrafu.
 - Máxima Verossimilhança: Programa PhymI.
- Implementar o sistema com suas principais funcionalidades na linguagem JAVA;
- Disponibilizar ao fim do projeto o instalador do *software* IGrafu 2.0 (Interface Gráfica do Grupo de Reconstrução de Árvores Filogenéticas da UESC) contendo todos os programas que necessita para ser executado.

1.4 Metodologia

Para o desenvolvimento da IGrafu 2.0 será adotada a metodologia de processo RUP (*Rational Unified Process*). (PAULA FILHO, 2003; SEABRA JÚNIOR, 2001), mais especificamente a metodologia iterativa, que consiste de uma fase de inspeção, escolha e desenvolvimento das iterações e, finalmente, a integração total da interface (LARMAN, 2007).

O software foi desenvolvido seguindo o paradigma de programação orientada a objetos (POO), sendo adotada para a implementação a linguagem de programação Java.

O documento de Requisitos de Software será escrito e modelado com a ajuda da IDE (*Integrated Development Environment*) para modelagem de dados JUDE (*Java and UML Developers' Environment*).

A codificação da IGrafu 2.0 contará com o auxílio da IDE (*Integrated Development Environment*) para desenvolvimento Eclipse, da API (*Application Programming Interface*) a linguagem utilizada será JAVA da Sun Microsystems.

1.5 Cronograma de Execução

Tarefa	Agosto	Setembro	Outubro	Novembro	Dezembro
Elaboração do pré-projeto de Estágio	X				
Fase de inspeção		X			
Fase de desenvolvimento das iterações		X	X	X	
Integração e Teste total da IGrafu 2.0				X	X
Elaboração do relatório de estágio curricular	X	X	X	X	X

Tabela 1 – Cronograma de execução

1.6 Justificativa e Contribuições

O *software* a ser desenvolvido será projetado e dividido de acordo com as solicitações da orientadora deste projeto Martha Ximena, a qual possui vasta experiência na área de bioinformática.

A utilização dos programas a serem incorporados à IGrafu através de uma interface gráfica amigável, totalmente transparente para o usuário e de fácil utilização, tanto para a entrada de dados quanto para visualização dos resultados gerados ajudará muito os profissionais que os utilizam, pois estes são não-especialistas em computação, o que muitas vezes dificulta o seu uso dos programas em modo texto (console Linux).

1.7 Organização Deste Relatório

Este relatório dividi-se da seguinte forma:

- Capítulo 1 – Introdução: Faz o leitor ter uma visão geral sobre o estágio, o problema abordado, os objetivos, um resumo da metodologia aplicada e a justificativa de o fazê-lo.
- Capítulo 2 – Fundamentos Teóricos: Aqui temos em profundidade os conceitos determinantes para a realização da tarefa a qual o estágio se propôs a cumprir, como bioinformática, métodos filogenéticos, árvores filogenéticas e *bootstrap*.
- Capítulo 3 – Ferramentas e Métodos: Guia de todas as ferramentas usadas no processo e de todos os métodos aplicados.
- Capítulo 4 – Composição do *Software*: Possui uma lista de programas que compõem a IGrifu e uma breve descrição de cada um.
- Capítulo 5 – Desenvolvimento: Explica toda fase de desenvolvimento do *software*.
- Capítulo 6 – Testes: Mostra como ocorreram os testes da IGrifu 2.0.
- Capítulo 7 – Resultados e Conclusões: Todas as observações do autor a cerca do processo e suas conclusões.
- Capítulo 8 – Trabalhos Futuros: Expectativas futuras para continuidade deste projeto.
- Referências – Ligação do texto as suas fontes.
- Apêndice 1 – Especificação de Requisitos de *Software*.
- Apêndice 2 – Telas do Sistema e Controles de Navegação.

2 FUNDAMENTOS TEÓRICOS

2.1 Bioinformática

A bioinformática (ou biocomputação) é a combinação dos conhecimentos químicos, físicos, biológicos, da engenharia genética e da ciência da computação, ela consiste na utilização dos meios computacionais para analisar as moléculas (principalmente de DNA), combinando com modelos matemáticos e utilizando também a estatística para interpretar e analisar os problemas biológicos em questão.

Para que tenha um resultado satisfatório, a bioinformática integra essencialmente à Engenharia de Software, já que é necessário o desenvolvimento de softwares para tratar todos dados já existentes e identificar os genes, além disso: prevêm a configuração tridimensional das proteínas, na identificação dos inibidores de enzimas, na simulação de células, agrupamento de proteínas, na montagem de árvores filogenéticas e analisar todos os experimentos da expressão gênica.

A partir dos projetos genoma, a bioinformática começou a se destacar, na década de 1990, produzindo volumes intensos de informações, ela utilizou a ajuda da internet para compartilhar dados e também do desenvolvimento da Informática (Computação), em relação à produção de processadores mais rápidos e também mais memórias, fatores que foram decisivos nos avanços e consolidação da bioinformática no meio científico.

Segundo uma pesquisadora do LNCC (laboratório Nacional de Computação e Estatística), “A Bioinformática é uma das áreas que mais vai crescer, pois é uma área multidisciplinar que vai formar um profissional com essa dupla visão.”.

2.2 Métodos Filogenéticos

2.2.1 Máxima Verossimilhança

O método de MV baseia-se na reconstrução filogenética através da busca por uma árvore que maximize a probabilidade dos dados observados. Neste sentido, o método de MV calcula as probabilidades associadas a diferentes topologias e cada uma delas com as variações nos tamanhos dos ramos, considerando o modelo evolutivo escolhido.

2.2.2 Distância

O método de Distância funciona basicamente em dois passos, sendo que o primeiro deles é a redução das variações entre seqüências alinhadas a valores de distância dispostos em uma matriz. No segundo passo, estes valores são utilizados na reconstrução filogenética. Um dos métodos de distância mais comuns é a chamada distância p , que expressa o número de sítios variáveis entre duas seqüências com relação ao total de sítios comparados.

2.3 Árvore Filogenética

Uma árvore filogenética ou cladograma, por vezes também designada por Árvore da Vida, é uma exibição em forma de uma árvore das relações evolutivas entre várias espécies ou outras entidades que podem ter um antepassado em comum. Em uma árvore filogenética, cada nodo com descendentes representa o mais recente antepassado comum, e os comprimentos dos ramos podem representar estimativas do tempo evolutivo. Cada nodo terminal em uma árvore filogenética é chamado uma unidade taxonômica. Nodos internos geralmente são chamados de Unidades Taxonômicas Hipotéticas. As árvores filogenéticas são confeccionadas a partir de uma matriz contendo os dados disponíveis (morfológicos, químicos ou genéticos) sobre os táxons estudados. Estes dados são comparados, e os táxons agrupados pelas semelhanças e diferenças entre si em clados¹. Na figura 1 abaixo temos um exemplo da representação gráfica de uma árvore filogenética.

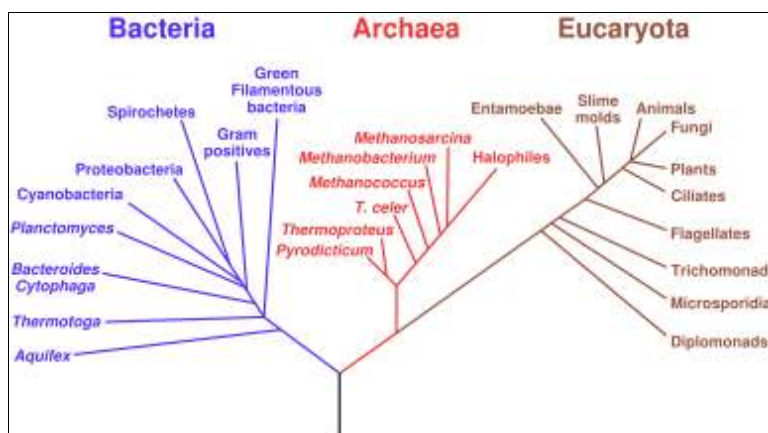


Figura 1 – Representação gráfica de uma árvore filogenética.

¹ Em cladística, um clado é um grupo de organismos originados de um único ancestral comum. Em biologia se chama clado cada um dos ramos da árvore filogenética. Por conseguinte um clado é um grupo de espécies com um ancestral comum.

2.4 *Bootstrap*

O *bootstrap* funciona gerando conjuntos modificados de dados, obtidos aleatoriamente a partir dos dados do alinhamento. Para cada conjunto aleatório de dados obtidos é estimada uma árvore. As novas árvores, geradas a partir dos conjuntos modificados dos dados de entrada, são comparadas. Cada um dos ramos da árvore final recebe então um valor de probabilidade, que é obtido do número de novas árvores onde esse ramo ocorreu dividido pelo número total de novas árvores estimadas. Probabilidades altas indicam que, mesmo com algumas alterações, os dados suportam o ramo ao qual essa probabilidade se refere e probabilidades baixas significam que, com a amostra analisada, não se pode ter certeza de que determinado ramo seja correto.

3 FERRAMENTAS E MÉTODOS

Para desenvolvimento deste relatório foram utilizados os seguintes métodos e ferramentas:

3.1 Rational Unified Process (RUP)

O RUP, também conhecido como Processo Unificado, é um processo de engenharia de *software* criado pela *Rational Software Corporation* que fornece técnicas a serem seguidas pelos membros da equipe de desenvolvimento de *software* com o objetivo de aumentar a produtividade.

O uso da orientação a objetos, o uso de modelos visuais com a notação UML e o desenvolvimento iterativo e incremental são as principais características do Processo Unificado.

3.2 Programação Orientada a Objetos (POO)

A programação orientada a objetos tem como principais objetivos reduzir a complexidade no desenvolvimento de *software* e aumentar sua produtividade. Podemos considerar que a POO é uma evolução de práticas que são recomendadas na programação estruturada, mas não formalizadas, como o uso de variáveis locais, visibilidade e escopo. O modelo de objetos permite a criação de bibliotecas que tornam efetivos o compartilhamento e a reutilização de código,

reduzindo o tempo de desenvolvimento e, principalmente, simplificando o processo de manutenção das aplicações.

Um objeto é uma abstração de *software* que pode representar algo real ou virtual. Um exemplo de objeto poderia ser um automóvel. O objeto automóvel possui propriedades (variáveis), como velocidade, número de portas e limite de passageiros. O objeto automóvel também possui procedimentos (métodos), como ligar, desligar, acelerar e parar.

Exemplos de linguagens de programação orientada a objetos são: Perl, Python, Ruby, Php, ColdFusion, C++, Object Pascal, Java, Javascript, ActionScript e C#.

3.3 Java

Desenvolvida na década de 90 pelo programador da *Sun Microsystems* James Gosling, a linguagem Java, diferentemente das linguagens convencionais, é compilada para uma representação intermediária chamada *bytecode*. Os *bytecodes* são interpretados pela Máquina Virtual Java (JVM ou *Java Virtual Machine*) que se responsabiliza em gerar as instruções ao *hardware* específico.

Java é uma linguagem de programação totalmente orientada a objetos, o que facilita a definição de interfaces de comunicação e aumenta a reusabilidade de código. Além de possuir sintaxe semelhante a C++, Java possui ferramentas que auxiliam na documentação Java, possui também uma vasta API com a definição de

classes e métodos capazes de propiciar o desenvolvimento do sistema de maneira simplificada, aumentando a legibilidade do código.

Visando tornar o trabalho portátil, flexível, legível, e robusto, será utilizada para a implementação deste trabalho a linguagem de programação Java, pois além de cumprir todos os requisitos exigidos, Java é totalmente portátil entre plataformas, possui uma vasta biblioteca de funções, é orientada a objetos e é uma linguagem familiar a este autor, o que elimina o tempo de adaptação á uma outra linguagem.

3.4 UML

A *Unified Modeling Language* (UML) é uma linguagem de modelagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objetos que permite que desenvolvedores visualizem os produtos e serviços de seu trabalho em diagramas padronizados.

Por meio de seus diagramas é possível representar sistemas de *software* sob diversas perspectivas de visualização, o que facilita a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema - gerentes, coordenadores, analistas, desenvolvedores, clientes - por apresentar um vocabulário de fácil entendimento.

3.5 JUDE

JUDE ou *Java and UML Developer Environment* é uma das ferramentas grátis para UML mais poderosas disponíveis atualmente. Com características que não são encontradas nas outras ferramentas grátis, como adicionar métodos no diagrama de seqüência e a alteração refletir no diagrama de classes.

JUDE fornece diversas facilidades para o desenvolvimento dos diagramas presentes na UML, possibilitando a exportação dos mesmos nos formatos JPG e PNG.

3.6 Eclipse

O projeto Eclipse.org é um consórcio de mais de 50 grandes empresas da área (como IBM, Borland, Suse, Intel, etc) que visa à criação de ferramentas gratuitas de desenvolvimento para Java. Um dos seus resultados é o ambiente de desenvolvimento Eclipse, o qual vem se tornando referência no mercado por ser poderoso, flexível, fácil de usar e livre. Com o Eclipse, é possível personalizar totalmente o ambiente de acordo com o projeto que está sendo desenvolvido, dos simples aos complexos, seja desenvolvimento para a plataforma *Web*, seja para *Desktop*. Conta, ainda, com suporte ao desenvolvimento em equipe e com centenas de *plug-ins* que atendem as necessidades dos programadores.

4 COMPOSIÇÃO DO SOFTWARE

A IGrafu 2.0 é composta pelos seguintes softwares:

4.1 Seqboot

Permite, a partir de seqüências primordiais, a produção de um grande número de *bootstrapped*.

4.2 Consense

Gera um arquivo contendo a árvore filogenética com maior probabilidade.

4.3 DiGrafu

Software que implementa o método de Distância.

4.4 PHYML

Software que utiliza o método de Máxima Verossimilhança.

4.5 Hypertree

Software utilizado para visualizar a árvore filogenética gerada.

5 DESENVOLVIMENTO

Este projeto foi construído seguindo uma metodologia de desenvolvimento de *software* própria, organizada em fases de inspeção, escolha e desenvolvimento das iterações e, finalmente, a integração total da interface dentro de um ciclo de vida baseado no modelo iterativo de desenvolvimento de *software*.

5.1 Elaboração

Foram levantados os principais requisitos através da interação com a orientadora deste projeto, o que tornou possível a definição do escopo do projeto. Os requisitos foram retratados através de texto, diagramas e modelos computacionais de representação utilizando as técnicas de orientação a objetos e diagramas da UML.

A Figura 2, abaixo, traz o diagrama de casos de uso do sistema que descreve os aspectos fundamentais do domínio de aplicação. O estudo dos casos de uso do sistema será feito de forma mais detalhada no Anexo 1 – Especificação de Requisitos de Software, deste relatório.

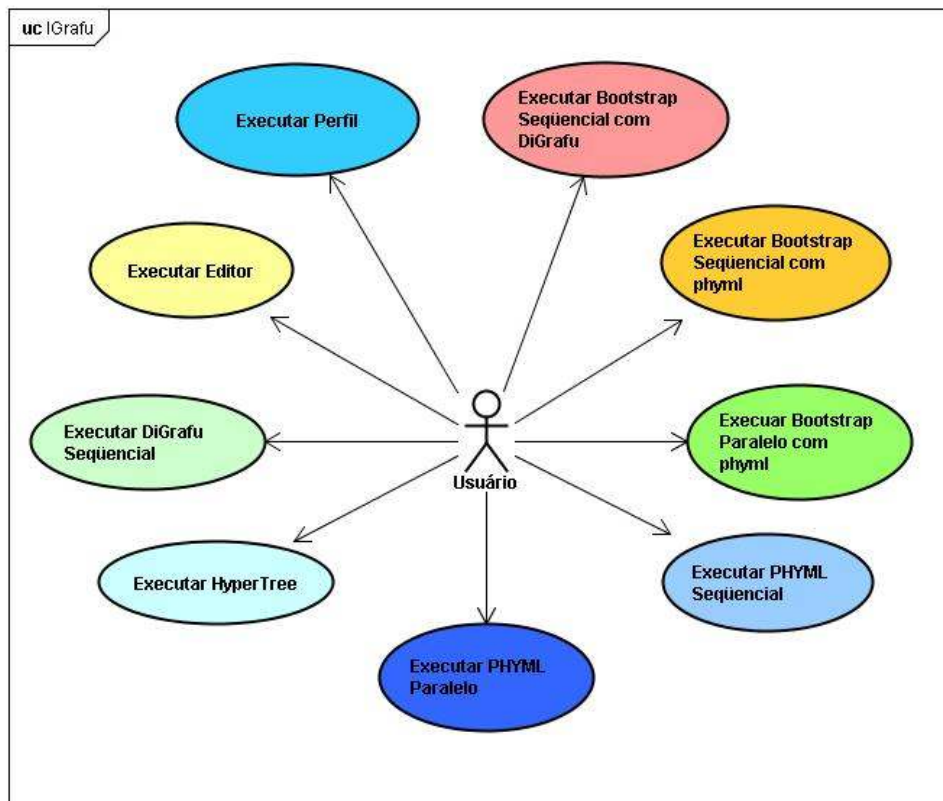


Figura 2 – Casos de uso da IGRAFU.

Com os dados para o desenvolvimento do projeto coletados e o escopo definido, prosseguiu-se com a modelagem dos dados do sistema construída para representar o problema da forma mais legível possível. A Figura 3 exibe o diagrama de estados de navegação, o qual indica quais janelas compõem o sistema e quais eventos permitem ao usuário navegar de uma para outra. O anexo 2 – Telas do Sistema e Controles de Navegação, deste relatório, traz um maior detalhamento das janelas do sistema e suas relações.

5.2 Implementação do Projeto

Definido os componentes do sistema, como mostrado na Figura 3, iniciou-se a implementação dos componentes com suas funcionalidades específicas apresentadas a seguir:

1. IGRAFU – Splash: SplashWindow - Tela de inicialização do sistema, com o logotipo da IGRAFU e patrocinadores.
2. IGRAFU – Modo de Execução: ModoExecucao - Tela onde o usuário indica o modo de execução da IGrafu, informa o arquivo contendo a seqüência genética a ser processada e escolhe o idioma que deseja visualizar (inglês ou português(default)).
3. IGRAFU – Bootstrap: ModoManualBootstrap - Tela onde o usuário informa se deseja ou não utilizar bootstrap.
4. IGRAFU – Seqboot: Seqboot – Tela onde o usuário informa todos os dados de bootstrap (Seqboot e Consense) a serem processados.
5. IGRAFU – Métodos: Metodos – Tela onde o usuário escolhe qual método deseja utilizar. Os métodos disponíveis são: Máxima Verossimilhança e Distância.
6. IGRAFU – PhymI: PhymI – Tela onde o usuário informa todos os dados necessários para execução do PhymI.
7. IGRAFU – DIGRAFU: DiGrafu - Tela onde o usuário informa todos os dados necessários para execução do DIGRAFU.
8. HyperTree – Software utilizado para visualização de árvores filogenéticas.
9. Editor – Utilizado para abrir e gerar arquivos textos. Sua principal finalidade é que o usuário possa visualizar os arquivos gerados após a execução da IGrafu e gerar os arquivos de entrada do Seqboot, Consense, PhymI e DiGrafu.

As classes do sistema IGrAFU 2.0 foram organizadas em pacotes de acordo com o padrão MVC², a Figura 4 exemplifica esta estrutura de pacotes.

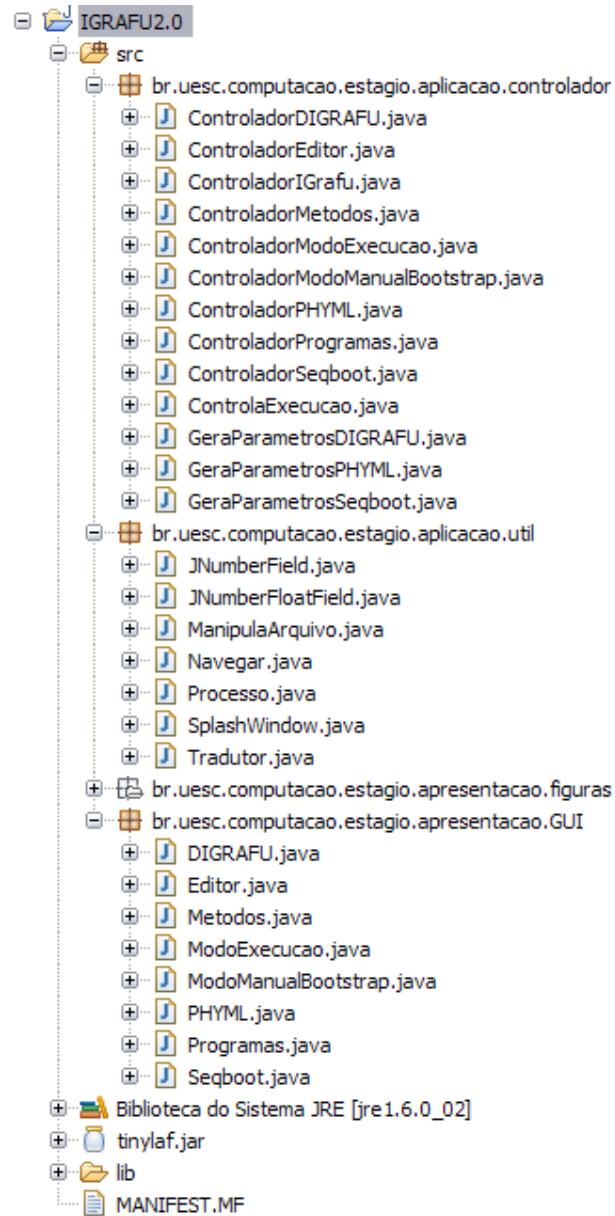


Figura 4 – Estrutura de pacotes do Eclipse.

² *Model View Controller* é um padrão de arquitetura de aplicações que visa separar a lógica da aplicação (*Model*), da interface do usuário (*View*) e do fluxo da aplicação (*Controller*).

Segundo o modelo proposto por Wazlawick (p. 257, 2004), a camada de interface foi dividida em duas subcamadas como mostrado na figura 4:

a) Apresentação: Uma camada com as classes que representam os objetos gráficos de interface e cujas responsabilidades se resumem a receber dados e comandos do usuário e apresentar resultados a ele. Camada composta pelos pacotes:

- `br.uesc.computacao.estagio.apresentacao.GUI`;
- `br.uesc.computacao.estagio.apresentacao.figuras`

b) Aplicação: Uma camada que controla a lógica de interface, abrindo e fechando janelas, habilitando e desabilitando botões etc. Camada composta pelos pacotes:

- `br.uesc.computacao.estagio.aplicacao.controlador`;
- `br.uesc.computacao.estagio.aplicacao.util`;

Sendo a IGrafu um *software* bastante modular, onde as classes da camada de Apresentação independem umas das outras, assim, todas tornam-se importantes para o sistema. O principal controlador do sistema é a classe `ControladorIGrafu`, que possui objetos para todas as outras classes controladoras além de possuir o método `main`, que determina o ponto de início de execução de qualquer aplicação Java. A Figura 5 mostra o diagrama de classe do pacote `br.uesc.computacao.estagio.aplicacao.controlador`.

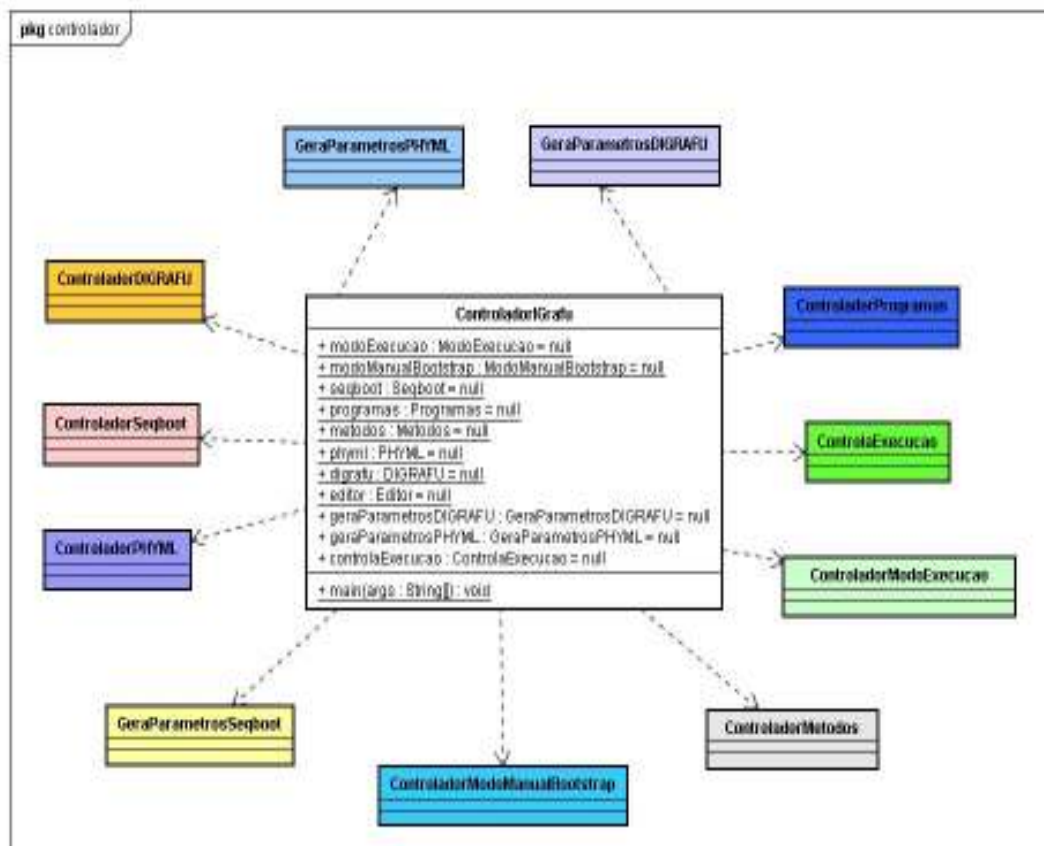


Figura 5 – Diagrama de classe do pacote br.uesc.computacao.estagio.aplicao.controlador.

6 TESTES

Os testes de execução da IGrafu 2.0 foram realizados no cluster do NBCGIB. O sistema se comportou dentro do previsto, com todas suas funcionalidades executando perfeitamente. Desta forma, considera-se a IGrafu 2.0 apta a desempenhar suas funções dentro da bioinformática.

7 RESULTADOS E CONCLUSÕES

Com desenvolvimento da interface gráfica integradora de soluções para reconstrução de árvores filogenéticas – IGrafu, foi possível obter uma ferramenta amigável, totalmente transparente para o usuário e de fácil utilização, tanto para a entrada de dados quanto para visualização dos resultados gerados, a qual disponibiliza os métodos de Distância e Máxima Verossimilhança. A IGrafu é uma interface gráfica auto contida, ou seja, trás em seu pacote de instalação todos os programas que utiliza, além de carrega todos os módulos necessários para seu funcionamento. Possui uma característica inovadora, onde a IGrafu pode ser utilizada em plataformas paralelas baseadas em clusters de PCs, sendo que possui opção para ser executada de maneira seqüencial em qualquer nó ou ser executada em paralelo, dando opção ao usuário escolher a quantidade de nós a serem utilizados, além de se adequar a ambiente desktop, onde sua execução é seqüencial. A IGrafu torna-se assim uma importante ferramenta para estudos e pesquisas relacionados a Genética Molecular e Bioinformática.

8 TRABALHOS FUTUROS

Os objetivos desta fase de desenvolvimento foram alcançados, mas o desenvolvimento da IGrafu continuará com a implementação de uma versão para Windows, inserção do modo de execução automático e de novos programas e métodos filogenéticos como:

Método:

- Bayes: Programa:
 - Mr. Bayes

- Máxima Verossimilhança: Programa:
 - TreeFinder

- Parcimonia:Programas:
 - NONA;

 - DNAPars;

O desenvolvimento continuará fazendo uso das tecnologias aqui apresentadas, pois estas se mostraram eficazes no desenvolvimento de sistemas dessa natureza.

REFERÊNCIAS

SCHADT, Eric; SINSHEIMER, Janet; LANGE, Kenneth. Computational Advances in Maximum Likelihood Methods for Molecular Phylogeny. In: **Genome Research**, 222-233, V. 8, 1998.

SCHMIDT, H. A. **Phylogenetic trees form large datasets**. Ph.D thesis. Dusseldorf University, 2003.

GUINDON, Stephane; GASCUEL, Oliver. A Simple, Fast, and Accurate Algorithm to Estimate Large Phylogenies by Maximum Likelihood. In: **Syst. Biol**, 693-704, V. 52, N° 5, 2003.

LARMAN, Craing. **Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientado a objetos e ao desenvolvimento iterativo**. 3. ed. Porto Alegre: Bookman, 2007.

DEITEL, H. M.; DEITEL, P.J. **Java - Como Programar**. 6. ed. Tradução de Edson Furmankiewicz. Porto Alegre: Bookman, 2001.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software: Fundamentos, Métodos e Padrões**. 2 ed. LTC, 2003.

SANDERSON, Michael; DRISKELL, Amy. The Challenge of constructing large phylogenetic trees. In: **Trends in Plant Science**, 374-379 V. 8, N° 8, 2003.

WAZLAWICK, Raul Sidnei. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. 4 ed. Rio de Janeiro: Campus, 2004.

SEABRA JÚNIOR, Rodolfo Moacir. **Análise e Projeto Orientado a Objetos usando UML e o Processo Unificado**. Belém, 2001. Disponível em: <http://www2.ufpa.br/cdesouza/teaching/cedai/APOOUMLP.pdf>

Maia, L.P; Morelli, E. “**Programação Orientada a Objetos**“. Disponível em <http://www.training.com.br/lpmaia/pub_prog_oo.htm>. Acesso em: 04/11/2007.

“**Seqboot -- Bootstrap, Jackknife, or Permutation Resampling of Molecular Sequence, Restriction Site, Gene Frequency or Character Data**”.

Disponível em: <<http://evolution.genetics.washington.edu/phylip/doc/seqboot.html>> Acesso em: 27/11/2007.

“**Árvore Filogenética**“. Disponível em:

<http://pt.wikipedia.org/wiki/%C3%81rvore_filogen%C3%A9tica>. Acesso em: 27/11/2007.

“**Jude ferramenta UML grátis**“. Disponível em <<http://jude.change-vision.com/jude-web/product/community.html>>. Acesso em 04/11/2007.

APÊNDICE 1 – Especificação de Requisitos de *Software*

1. INTRODUÇÃO

Este documento apresentará um catálogo com os principais requisitos do sistema IGrafu 2.0. Será apresentada também a especificação dos casos de uso que estão apresentados no catálogo.

O diagrama de Casos de Uso está contido.

2. CATÁLOGO DE CASOS DE USO

Número	Nome	Descrição
1.	Executar Bootstrap Seqüencial com DiGrafu	Usuário informa todos os parâmetros do <i>seqboot</i> , <i>consense</i> e DiGrafu necessários e executa as opções escolhidas seqüencialmente.
2.	Executar Bootstrap Seqüencial com phymI	Usuário informa todos os parâmetros do <i>seqboot</i> , <i>consense</i> e phymI necessários e executa as opções escolhidas seqüencialmente.
3.	Executar Bootstrap Paralelo com phymI	Usuário informa todos os parâmetros necessários e executa as opções escolhidas paralelamente.
4.	Executar phymI Seqüencial	Usuário informa todos os parâmetros necessários e executa o phymI Seqüencialmente.
5.	Executar phymI Paralelo	Usuário informa todos os parâmetros necessários e executa o phymI paralelamente
6.	Executar DiGrafu Seqüencial	Usuário informa todos os parâmetros necessários e executa o DiGrafu seqüencialmente.
7.	Executar Hypertree	Usuário visualiza a árvore filogenética gerada após a execução dos programas.
8.	Executar Editor	Usuário cria, edita ou gera texto (.txt) de entrada ou saída utilizados na execução da IGrafu.
9.	Executar Perfil	Usuário executa um perfil existente.

Tabela 2 - Casos de uso da IGrafu 2.0

3. DIAGRAMAS DO MODELO DE CASOS DE USO

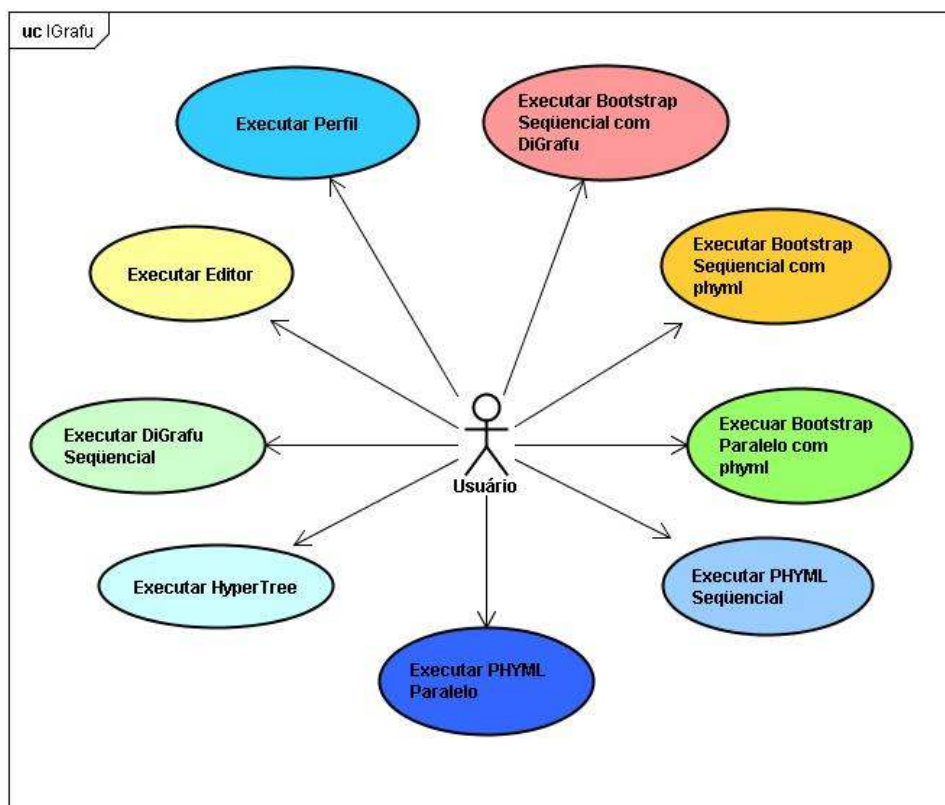


Figura 6 - Casos de uso do sistema.

4. ESPECIFICAÇÃO DOS CASOS DE USO

4.1 Caso de Uso:	Executar Bootstrap Seqüencial com DiGrafu
Atores:	Usuários do sistema
Descrição:	Usuário informa todos os parâmetros do <i>seqboot</i> e <i>consense</i> necessários, escolhe um programa (DiGrafu ou <i>phymI</i>) para gerar a árvore filogenética e executa as opções escolhidas seqüencialmente.

Seqüência Eventos

Ator	Sistema
1 - Este caso de uso começa quando o usuário executa o sistema.	2 - O sistema é carregado exibindo inicialmente a tela de <i>Splash</i> do sistema durante 5 segundos " <i>SplashWindow</i> " e logo em seguida carrega o sistema na tela Modo de Execução " <i>ModoExecuca</i> o" para escolha do modo de execução e arquivo da seqüência a ser utilizado. O idioma default é o português.
3 - O usuário escolhe o idioma a ser utilizado clicando no menu " <i>Configurações</i> ", depois no menu " <i>Idioma</i> " e escolhe o idioma inglês ou português.	4 - O sistema muda a linguagem da IGrifu para o idioma escolhido pelo usuário.
5 - O usuário escolhe o arquivo de entrada (.txt) clicando no botão "Abrir Arquivo", escolhe o modo de Execução "Manual" e clica no botão "Avançar".	6 - O sistema guarda o caminho onde está o arquivo da seqüência e abre a próxima tela "Bootstrap".
7 - O usuário escolhe a opção "Sim" e clica no botão "Avançar".	8 - O sistema abre a próxima tela "Seqboot & Consense", carregando todos os parâmetros destes programas.
9 - O usuário escolhe todos os parâmetros necessário para execução do <i>seqboot</i> e <i>consense</i> e clica no botão "Avançar".	10 - O sistema carrega a próxima tela "Métodos", com os métodos de Distância e Máxima Verossimilhança.
11 - O usuário informa qual método deseja utilizar. Caso escolha "Distância" e depois clique no botão "Avançar".	12 - O sistema abrirá a tela "Programas", exibindo o(s) programa(s) disponíveis referente ao método de Distância.
13 - O usuário escolhe o programa que deseja. Clicando em DiGrafu e depois no botão "Avançar".	14 - O sistema carregará a tela do DiGrafu, com todas suas funcionalidades.

- 15 - Usuário deve escolher as opções de execução do DiGrafu e clicar no botão “Executar” .
- 16 - O sistema abrirá uma caixa de diálogo perguntando onde deseja executar o DiGrafu – servidor (caso a IGrafu esteja instalada em um cluster será executado no nó principal, caso esteja em Pc Desktop, esta é a opção que deve ser escolhida), no1, no2, no3, no4.
- 17 - O usuário escolhe o local de execução e clica em “Executar”.
- 18 - O sistema executa as opções de bootstrap e o DiGrafu seqüencialmente no local indicado. Exibindo ao fim da execução uma mensagem que a execução foi bem sucedida ou caso tenha havido algum erro uma mensagem que ocorreu algum problema na execução.

4.2 Caso de Uso:	Executar Bootstrap Seqüencial com phymI
Atores:	Usuários do sistema
Descrição:	Usuário informa todos os parâmetros do <i>seqboot</i> , <i>consense</i> e phymI necessários e executa as opções escolhidas seqüencialmente.

Seqüência Eventos

Ator	Sistema
1 - Este caso de uso começa quando o usuário executa o sistema.	2 - O sistema é carregado exibindo inicialmente a tela de <i>Splash</i> do sistema durante 5 segundos “ <i>SplashWindow</i> ” e logo em seguida carrega o sistema na tela Modo de Execução “ <i>ModoExecucao</i> ” para escolha do modo de execução e arquivo da seqüência a ser utilizado. O idioma default é o português.
3 - O usuário escolhe o idioma a ser utilizado clicando no menu “ <i>Configurações</i> ”, depois no menu “ <i>Idioma</i> ” e escolhe o idioma inglês ou português.	4 - O sistema muda a linguagem da IGrafu para o idioma escolhido pelo usuário.

- 5 - O usuário escolhe o arquivo de entrada (.txt) clicando no botão “Abrir Arquivo”, escolhe o modo de Execução “Manual” e clica no botão “Avançar”.
- 6 - O sistema guarda o caminho onde está o arquivo da seqüência e abre a próxima tela “Bootstrap”.
- 7 - O usuário escolhe a opção “Sim” e clica no botão “Avançar”.
- 8 - O sistema abre a próxima tela “Seqboot & Consense”, carregando todos os parâmetros destes programas.
- 9 - O usuário escolhe todos os parâmetros necessário para execução do *seqboot* e *consense* e clica no botão “Avançar”.
- 10 - O sistema carrega a próxima tela “Métodos”, com os métodos de Distância e Máxima Verossimilhança.
- 11 - O usuário informa qual método deseja utilizar. Caso escolha “Maxima Verossimilhança” e depois clique no botão “Avançar”.
- 12 - O sistema abrirá a tela “Programas”, exibindo o(s) programa(s) disponíveis referente ao método de Máxima Verossimilhança.
- 13 - O usuário escolhe o programa que deseja. Clicando em phym1 e depois no botão “Avançar”.
- 14 - O sistema carregará a tela do phym1, com todas suas funcionalidades.
- 15 - Usuário deve escolher as opções de execução do phym1 e clicar no botão “Executar” .
- 16 - O sistema abrirá uma caixa de diálogo “Tipo de Execução”, perguntando o tipo de execução do phym1, “Paralelo” ou “Seqüencial” – servidor (caso a IGratu esteja instalada em um cluster será executado no nó principal, caso esteja em Pc Desktop, esta é a opção que deve ser escolhida), no1, no2, no3, no4.
- 17 - O usuário escolhe a aba “Seqüencial”, o local de execução e clica no botão “Executar”.
- 18 - O sistema executa as opções de bootstrap e o phym1 seqüencialmente no local indicado. Exibindo ao fim da execução uma mensagem que a execução foi bem sucedida ou caso tenha havido algum erro uma mensagem que ocorreu algum problema na execução.

4.3 Caso de Uso: Executar Bootstrap Paralelo com phym1

Atores:	Usuários do sistema
Descrição:	Usuário informa todos os parâmetros necessários e executa as opções escolhidas paralelamente.

Seqüência Eventos

Ator	Sistema
1 - Este caso de uso começa quando o usuário executa o sistema.	2 - O sistema é carregado exibindo inicialmente a tela de <i>Splash</i> do sistema durante 5 segundos " <i>SplashWindow</i> " e logo em seguida carrega o sistema na tela Modo de Execução " <i>ModoExecucao</i> " para escolha do modo de execução e arquivo da seqüência a ser utilizado. O idioma default é o português.
3 - O usuário escolhe o idioma a ser utilizado clicando no menu " <i>Configurações</i> ", depois no menu " <i>Idioma</i> " e escolhe o idioma inglês ou português.	4 - O sistema muda a linguagem da IGrafu para o idioma escolhido pelo usuário.
5 - O usuário escolhe o arquivo de entrada (.txt) clicando no botão "Abrir Arquivo", escolhe o modo de Execução "Manual" e clica no botão "Avançar".	6 - O sistema guarda o caminho onde está o arquivo da seqüência e abre a próxima tela "Bootstrap".
7 - O usuário escolhe a opção "Não" e clica no botão "Avançar".	8 - O sistema carrega a próxima tela "Métodos", com os métodos de Distância e Máxima Verossimilhança.
9 - O usuário informa qual método deseja utilizar. Caso escolha "Maxima Verossimilhança" e depois clique no botão "Avançar".	10 - O sistema abrirá a tela "Programas", exibindo o(s) programa(s) disponíveis referente ao método de Máxima Verossimilhança.
11 - O usuário escolhe o programa que deseja. Clicando em phym1 e depois no botão "Avançar".	12 - O sistema carregará a tela do phym1, com todas suas funcionalidades.
13 - Usuário deve escolher as opções de execução do phym1 e clicar no botão "Executar".	14 - O sistema abrirá uma caixa de diálogo "Tipo de Execução", perguntando o tipo de execução do phym1, "Paralelo" ou "Seqüencial" – servidor (caso a IGrafu esteja instalada em um cluster será executado no nó principal, caso esteja em Pc Desktop, esta é a opção que deve ser escolhida), no1, no2, no3, no4.

17 - O usuário escolhe a aba “Paralelo”, a quantidade de processadores que deseja utilizar e clica em “Executar”.

18 - O sistema executa as opções de bootstrap e o phymI paralelamente com a quantidade de processadores indicado. Exibindo ao fim da execução uma mensagem que a execução foi bem sucedida ou caso tenha havido algum erro uma mensagem que ocorreu algum problema na execução.

4.4 Caso de Uso:	Executar phymI Seqüencial
Atores:	Usuários do sistema
Descrição:	Usuário informa todos os parâmetros necessários e executa o phymI Seqüencialmente.

Seqüência Eventos

Ator	Sistema
1 - Este caso de uso começa quando o usuário executa o sistema.	2 - O sistema é carregado exibindo inicialmente a tela de <i>Splash</i> do sistema durante 5 segundos “ <i>SplashWindow</i> ” e logo em seguida carrega o sistema na tela Modo de Execução “ <i>ModoExecucao</i> ” para escolha do modo de execução e arquivo da seqüência a ser utilizado. O idioma default é o português.
3 - O usuário escolhe o idioma a ser utilizado clicando no menu “ <i>Configurações</i> ”, depois no menu “ <i>Idioma</i> ” e escolhe o idioma inglês ou português.	4 - O sistema muda a linguagem da IGratu para o idioma escolhido pelo usuário.
5 - O usuário escolhe o arquivo de entrada (.txt) clicando no botão “Abrir Arquivo”, escolhe o modo de Execução “Manual” e clica no botão “Avançar”.	6 - O sistema guarda o caminho onde está o arquivo da seqüência e abre a próxima tela “Bootstrap”.
7 - O usuário escolhe a opção “Não” e clica no botão “Avançar”.	8 - O sistema carrega a próxima tela “Métodos”, com os métodos de Distância e Máxima Verossimilhança.
9 - O usuário informa qual método deseja utilizar. Caso escolha “Maxima Verossimilhança” e depois clique no botão “Avançar”.	10 - O sistema abrirá a tela “Programas”, exibindo o(s) programa(s) disponíveis referente ao método de Máxima Verossimilhança.

11 – O usuário escolhe o programa que deseja. Clicando em phym1 e depois no botão “Avançar”.

13 - Usuário deve escolher as opções de execução do phym1 e clicar no botão “Executar”.

17 - O usuário escolhe a aba “Seqüencial”, o local onde deseja executar e clica em “Executar”.

12 - O sistema carregará a tela do phym1, com todas suas funcionalidades.

14 - O sistema abrirá uma caixa de diálogo “Tipo de Execução”, perguntando o tipo de execução do phym1, “Paralelo” ou “Seqüencial” – servidor (caso a IGrafu esteja instalada em um cluster será executado no nó principal, caso esteja em Pc Desktop, esta é a opção que deve ser escolhida), no1, no2, no3, no4.

18 - O sistema executa as opções do phym1 seqüencialmente no local indicado. Exibindo ao fim da execução uma mensagem que a execução foi bem sucedida ou caso tenha havido algum erro uma mensagem que ocorreu algum problema na execução.

4.5 Caso de Uso:	Executar phym1 Paralelo
Atores:	Usuários do sistema
Descrição:	Usuário informa todos os parâmetros necessários e executa o phym1 paralelamente

Seqüência Eventos

Ator	Sistema
1 - Este caso de uso começa quando o usuário executa o sistema.	2 - O sistema é carregado exibindo inicialmente a tela de <i>Splash</i> do sistema durante 5 segundos “ <i>SplashWindow</i> ” e logo em seguida carrega o sistema na tela Modo de Execução “ <i>ModoExecucao</i> ” para escolha do modo de execução e arquivo da seqüência a ser utilizado. O idioma default é o português.
3 - O usuário escolhe o idioma a ser utilizado clicando no menu “ <i>Configurações</i> ”, depois no menu “ <i>Idioma</i> ” e escolhe o idioma inglês ou português.	4 - O sistema muda a linguagem da IGrafu para o idioma escolhido pelo usuário.

- 5 - O usuário escolhe o arquivo de entrada (.txt) clicando no botão “Abrir Arquivo”, escolhe o modo de Execução “Manual” e clica no botão “Avançar”.
- 6 - O sistema guarda o caminho onde está o arquivo da seqüência e abre a próxima tela “Bootstrap”.
- 7 - O usuário escolhe a opção “Não” e clica no botão “Avançar”.
- 8 - O sistema carrega a próxima tela “Métodos”, com os métodos de Distância e Máxima Verossimilhança.
- 9 - O usuário informa qual método deseja utilizar. Caso escolha “Maxima Verossimilhança” e depois clique no botão “Avançar”.
- 10 - O sistema abrirá a tela “Programas”, exibindo o(s) programa(s) disponíveis referente ao método de Máxima Verossimilhança.
- 11 - O usuário escolhe o programa que deseja. Clicando em phym1 e depois no botão “Avançar”.
- 12 - O sistema carregará a tela do phym1, com todas suas funcionalidades.
- 13 - Usuário deve escolher as opções de execução do phym1 e clicar no botão “Executar”.
- 14 - O sistema abrirá uma caixa de diálogo “Tipo de Execução”, perguntando o tipo de execução do phym1, “Paralelo” ou “Seqüencial” – servidor (caso a IGrafu esteja instalada em um cluster será executado no nó principal, caso esteja em Pc Desktop, esta é a opção que deve ser escolhida), no1, no2, no3, no4.
- 17 - O usuário escolhe a aba “Paralelo”, a quantidade de processadores que deseja utilizar e clica em “Executar”.
- 18 - O sistema executa as opções d o phym1 paralelamente com a quantidade de processadores desejada. Exibindo ao fim da execução uma mensagem que a execução foi bem sucedida ou caso tenha havido algum erro uma mensagem que ocorreu algum problema na execução.

4.6 Caso de Uso:	Executar DiGrafu Seqüencial
Atores:	Usuários do sistema
Descrição:	Usuário informa todos os parâmetros necessários e executa o DiGrafu seqüencialmente.

Seqüência Eventos

Ator	Sistema
------	---------

- 1 - Este caso de uso começa quando o usuário executa o sistema.
- 2 - O sistema é carregado exibindo inicialmente a tela de *Splash* do sistema durante 5 segundos "*SplashWindow*" e logo em seguida carrega o sistema na tela Modo de Execução "*ModoExecucao*" para escolha do modo de execução e arquivo da seqüência a ser utilizado. O idioma default é o português.
- 3 - O usuário escolhe o idioma a ser utilizado clicando no menu "*Configurações*", depois no menu "*Idioma*" e escolhe o idioma inglês ou português.
- 4 - O sistema muda a linguagem da IGrafu para o idioma escolhido pelo usuário.
- 5 - O usuário escolhe o arquivo de entrada (.txt) clicando no botão "Abrir Arquivo", escolhe o modo de Execução "Manual" e clica no botão "Avançar".
- 6 - O sistema guarda o caminho onde está o arquivo da seqüência e abre a próxima tela "Bootstrap".
- 7 - O usuário escolhe a opção "Não" e clica no botão "Avançar".
- 8 - O sistema carrega a próxima tela "Métodos", com os métodos de Distância e Máxima Verossimilhança.
- 9 - O usuário informa qual método deseja utilizar. Caso escolha "Distância" e depois clique no botão "Avançar".
- 10 - O sistema abrirá a tela "Programas", exibindo o(s) programa(s) disponíveis referente ao método de Distância.
- 11 - O usuário escolhe o programa que deseja. Clicando em DiGrafu e depois no botão "Avançar".
- 12 - O sistema carregará a tela do DiGrafu, com todas suas funcionalidades.
- 13 - Usuário deve escolher as opções de execução do DiGrafu e clicar no botão "Executar".
- 14 - O sistema abrirá uma caixa de diálogo perguntando onde deseja executar o DiGrafu – servidor (caso a IGrafu esteja instalada em um cluster será executado no nó principal, caso esteja em Pc Desktop, esta é a opção que deve ser escolhida), no1, no2, no3, no4.
- 15 - O usuário escolhe o local onde deseja executar e clica em "Executar".
- 16 - O sistema executa as opções do DiGrafu seqüencialmente no local indicado. Exibindo ao fim da execução uma mensagem que a execução foi bem sucedida ou caso tenha havido algum erro uma mensagem que ocorreu algum problema na execução.

4.7 Caso de Uso: Executar Hypertree

Atores: Usuários do sistema

Descrição: Usuário visualiza a árvore filogenética gerada após a execução dos programas.

Seqüência Eventos

Ator	Sistema
1 - Esse caso de uso começa quando o usuário clica no botão “Visualizar” das telas phyml, DiGrafu ou Bootstrap	2 - Caso o PHYML ou DiGrafu já tenham sido executados e o arquivo de visualização da árvore filogenética tenha sido gerado corretamente o Hypertree iniciará mostrando a árvore filogenética gerada pelo PHYML ou DiGrafu. Caso o PHYML ou DiGrafu não tenham sido executados e/ou o arquivo de visualização da árvore filogenética não exista, o Hypertree iniciará sem abrir nenhuma árvore filogenética.

4.8 Caso de Uso: Executar Editor

Atores: Usuários do sistema

Descrição: Usuário cria, edita ou gera texto (.txt) de entrada ou saída utilizados na execução da IGrafu.

Seqüência Eventos

Ator	Sistema
1 - Esse caso de uso começa quando o usuário clica no botão “Editor” nas telas Seqboot ou PHYML ou DiGrafu.	2 - O sistema carrega a tela do Editor com todas suas funcionalidades.

4.9 Caso de Uso: Executar Perfil

Atores: Usuários do sistema

Descrição: Usuário executa um perfil existente

Seqüência Eventos

Ator	Sistema
1 - Esse caso de uso começa quando o usuário clica no botão " <i>Perfil</i> " na tela Modo de Execução	
2 - Usuário clica carrega a seqüência a ser utilizada clicando no botão "Carregar seqüência"	3 - O sistema carrega a seqüência selecionada.
4 - Usuário clica no botão " <i>Avançar</i> ".	5 - O sistema abre uma caixa de diálogo exibindo os perfis criados em outras execuções pelo usuário.
6 - Usuário escolhe o perfil a ser executado.	
7 - Usuário clica no botão " <i>Executar</i> "	8 - O sistema executa o perfil selecionado
	9 - O sistema exibe o botão " <i>Visualizar</i> ".
10 - Usuário clica no botão " <i>Visualizar</i> " para ver a árvore filogenética gerada.	

APÊNDICE 2 – Telas do Sistema e Controles de Navegação

1 INTRODUÇÃO

Este documento apresentará as telas que compõem a interface de usuário e os controles de navegação entre estas.

2 LISTA DAS INTERFACES DE USUÁRIO E AÇÕES RELACIONADAS

2.1 Lista das interfaces de usuário e respectivas ações relacionadas

Nº	Nome Interface	Ação relacionada
1	TELA_SPLASH	---
2	TELA_MODALIDADE	Escolher o modo de execução da IGratu, informa o arquivo de seqüência a ser utilizado na execução e escolher o idioma a ser utilizado.
3	TELA_BOOTSTRAP	Escolher se deseja utilizar <i>bootstrap</i> ou não.
4	TELA_BOOTSTRAP_SEQBOOT_SEQÜÊNCIA	Informar todos os dados do arquivo de entrada a ser passado para o <i>seqboot</i> .
5	TELA_BOOTSTRAP_SEQBOOT_PARÂMETROS	Informar os parâmetros adicionais de execução do <i>seqboot</i> .
6	TELA_BOOTSTRAP_CONSENSE	Informar os parâmetros de execução do <i>consense</i> .
7	TELA_MÉTODOS	Informar qual método deseja utilizar.
8	TELA_PROGRAMAS_MÁXIMA_VEROSSIMILHANÇA	Escolher qual programa de Máxima Verossimilhança deseja utilizar.
9	TELA_PROGRAMAS_DISTÂNCIA	Escolher qual programa de Distância deseja utilizar.
10	TELA_PHYML_SEQÜÊNCIA	Informar todos os dados referentes ao arquivo da seqüência passado para o

		phyml.
11	TELA_PHYML_MODELO_DNA	Informar qual modelo de DNA deseja utilizar e seus respectivos parâmetros.
12	TELA_PHYML_MODELO_AA	Informar qual modelo de AA deseja utilizar e seus respectivos parâmetros.
13	TELA_PHYML_ÁRVORE	Informar os dados referente à árvore filogenética a ser gerada.
14	TELA_PHYML_TIPO_EXECUÇÃO	Informar o tipo de execução: Seqüencial ou Paralelo. Caso seqüencial, informar em qual nó do cluster deseja executar. Caso Paralelo, informar a quantidade de processadores que deseja utilizar.
16	TELA_DIGRAFU_SEQÜÊNCIA	Informar todos os dados referente ao arquivo da seqüência passado para o DiGrafu.
16	TELA_DIGRAFU_MODELO_DNA	Informar qual modelo de DNA deseja utilizar e seus respectivos parâmetros.
17	TELA_DIGRAFU_LOCAL_EXECUÇÃO	Informar o local de execução, em qual nó do cluster deseja executar.
18	TELA_EDITOR	Criar, editar ou gerar texto (.txt) de entrada ou saída utilizados na execução da IGrafu.
19	TELA_HYPERTREE	Visualizar a árvore filogenética gerada após a execução dos programas.

Tabela 3 – Interface de usuário e suas respectivas ações.

2.2 Telas da IGrafu 2.0

2.2.1 TELA_SPLASH



Figura 7 - Tela de Splash.

Descrição: Tela de abertura da IGrafu 2.0.

2.2.2 TELA_MODO_EXECUÇÃO

Campo	Ação
Botão “Avançar”	Abre a tela “Bootstrap”.

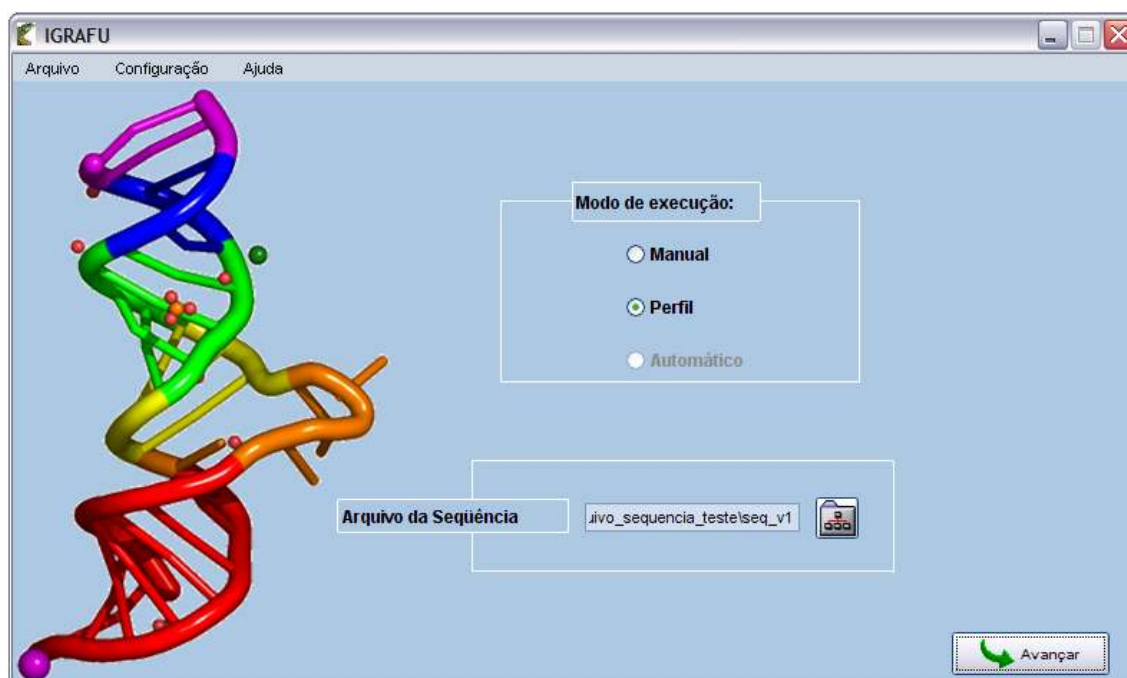


Figura 8 – Modo de execução.

Descrição: Local onde o usuário escolhe o modo de execução da IGrifu, informa o arquivo de seqüência a ser utilizado na execução e escolher o idioma a ser utilizado.

2.2.3 TELA_BOOTSTRAP

Campo	Ação
RadioButton “Sim” + Botão “Avançar”	Abre a próxima tela “Bootstrap” – Parâmetros <i>seqboot</i> e <i>consense</i> .
RadioButton “Não” + Botão “Avançar”	Abre a próxima tela “Métodos”.
Botão “Voltar”	Abre a tela anterior “Bootstrap”.

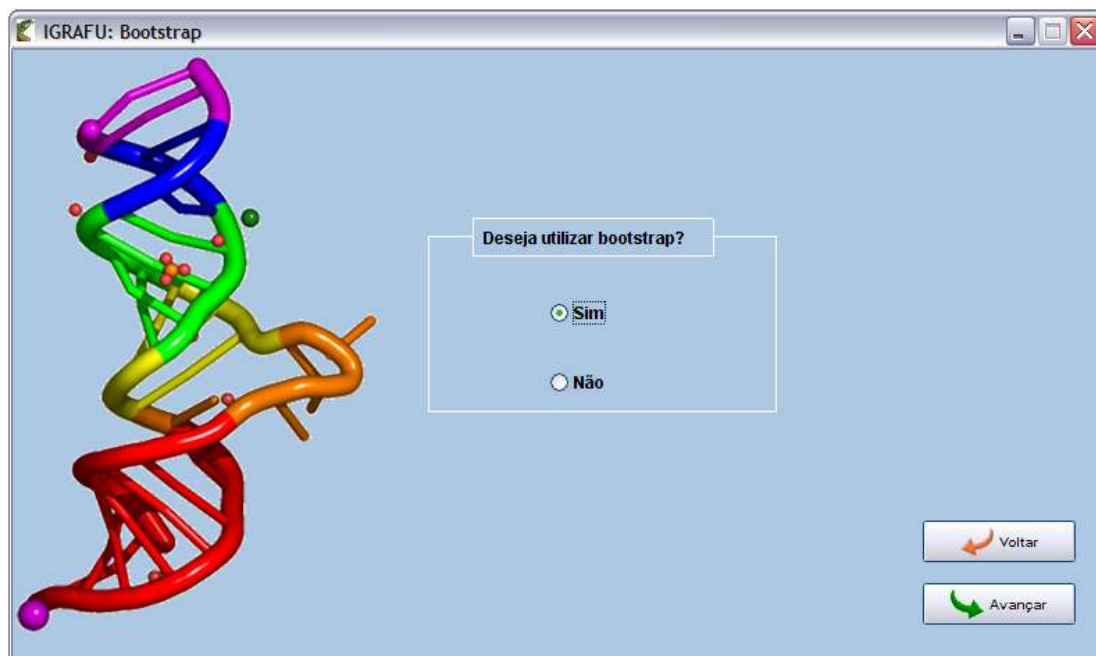


Figura 9 - Tela de escolha do bootstrap.

Descrição: Tela onde o usuário escolhe se deseja utilizar *bootstrap* ou não.

2.2.4 TELA_BOOTSTRAP_SEQBOOT_SEQÜÊNCIA

Campo	Ação
Botão “Editor”	Abre a tela “Editor”.
Botão “Avançar”	Abre a próxima tela “Métodos”.
Botão “Voltar”	Abre a tela anterior “Bootstrap”.

Figura 10 - Tela do seqboot – Seqüência.

Descrição: Tela onde o usuário informa todos os dados do arquivo de entrada a ser passado para o *seqboot*.

2.2.5 TELA_BOOTSTRAP_SEQBOOT_PARÂMETROS

Campo	Ação
Botão “Editor”	Abre a tela “Editor”.
Botão “Avançar”	Abre a próxima tela “Métodos”.
Botão “Voltar”	Abre a tela anterior “Bootstrap”.

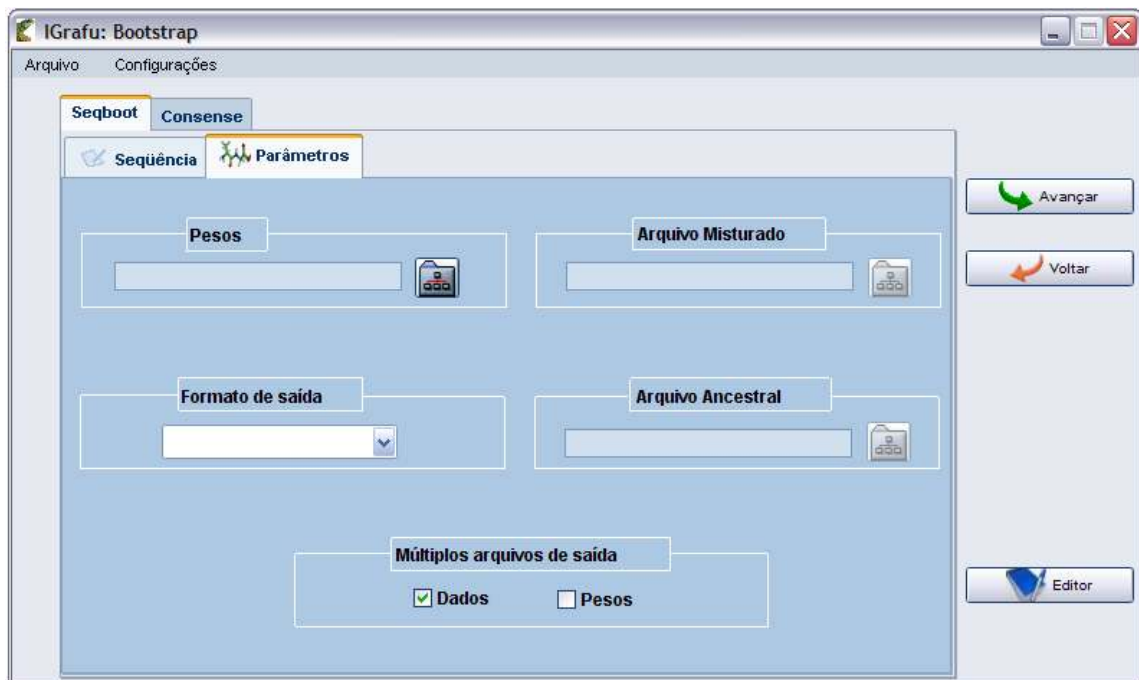


Figura 11 - Tela seqboot – Parâmetros.

Descrição: Tela onde o usuário informa os parâmetros adicionais de execução do *seqboot*.

2.2.6 TELA_BOOTSTRAP_CONSENSE

Campo	Ação
Botão “Editor”	Abre a tela “Editor”.
Botão “Avançar”	Abre a próxima tela “Métodos”.
Botão “Voltar”	Abre a tela anterior “Bootstrap”.

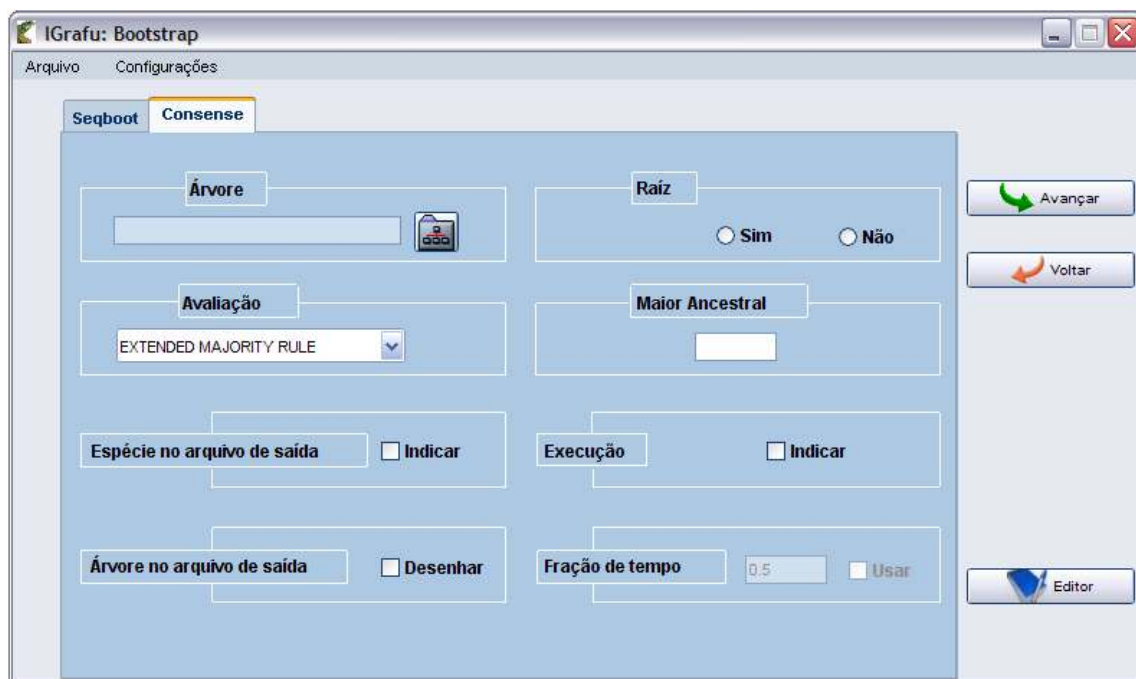


Figura 12 - Tela Consense.

Descrição: Tela onde o usuário informa os parâmetros de execução do *consense*.

2.2.7 TELA_MÉTODOS

Campo	Ação
Botão “Avançar”	Abre a próxima tela “Programas”.
Botão “Voltar”	Abre a tela anterior “Bootstrap”.

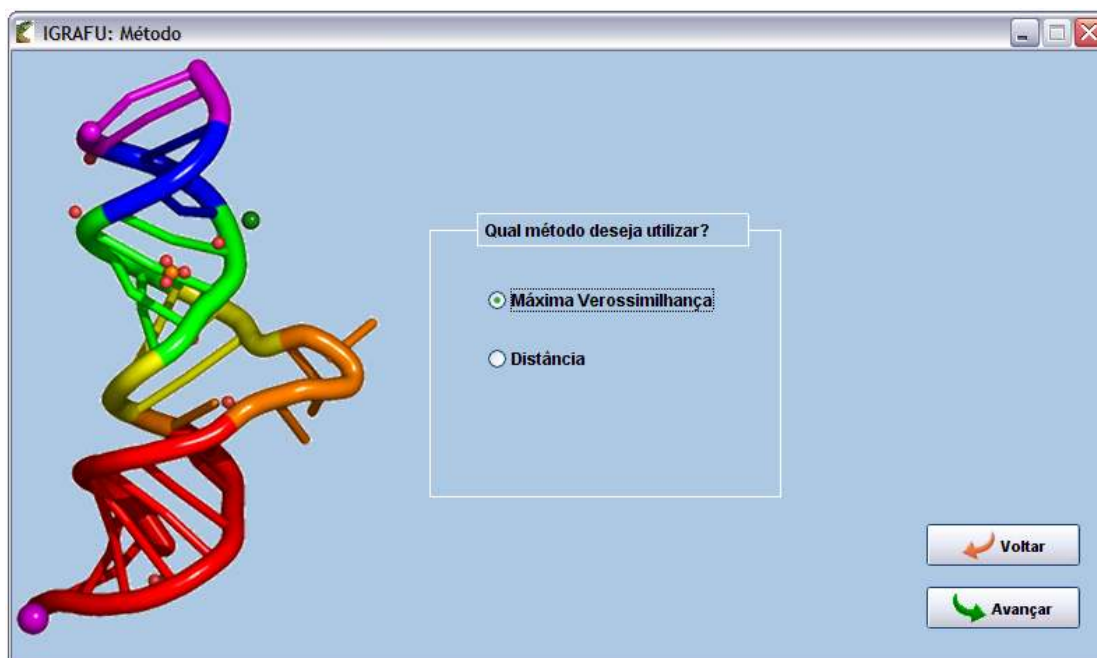


Figura 13 - Tela Metodos.

Descrição: Tela onde o usuário informa qual método deseja utilizar.

2.2.8 TELA_PROGRAMAS_MÁXIMA_VEROSSIMILHANÇA

Campo	Ação
Botão “Avançar”	Abre a próxima tela “PHYML”.
Botão “Voltar”	Abre a tela anterior “Métodos”.

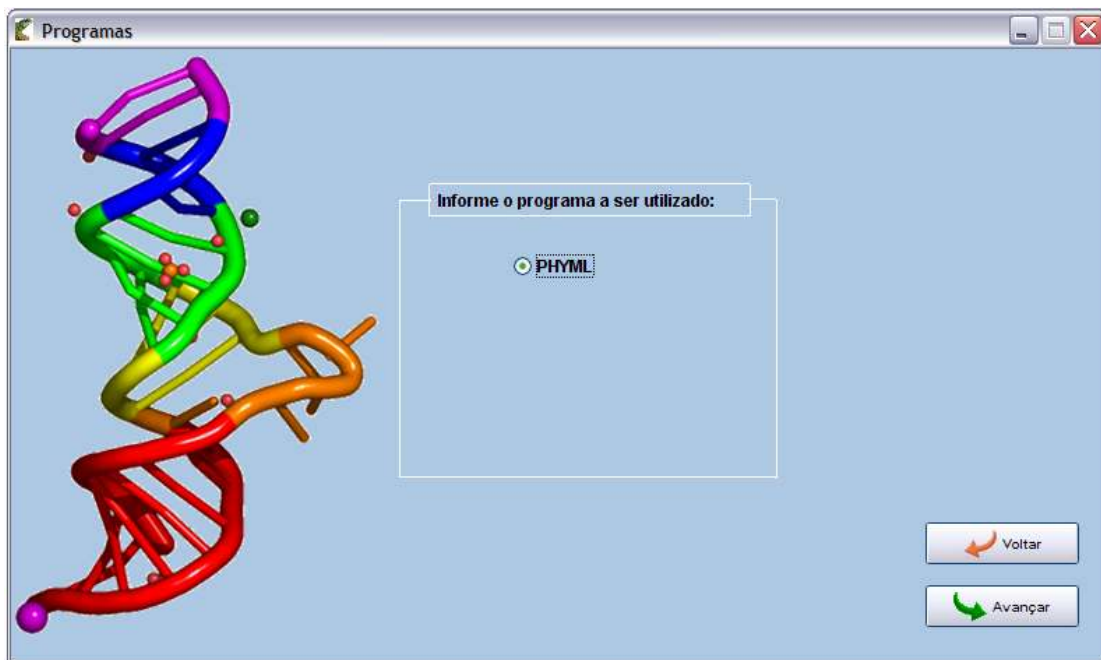


Figura 14 - Tela Programas - MV.

Descrição: Tela onde o usuário escolhe qual programa de Máxima Verossimilhança deseja utilizar.

2.2.9 TELA_PROGRAMAS_DISTÂNCIA

Campo	Ação
Botão “Avançar”	Abre a próxima tela “DiGrafu”.
Botão “Voltar”	Abre a tela anterior “Métodos”.

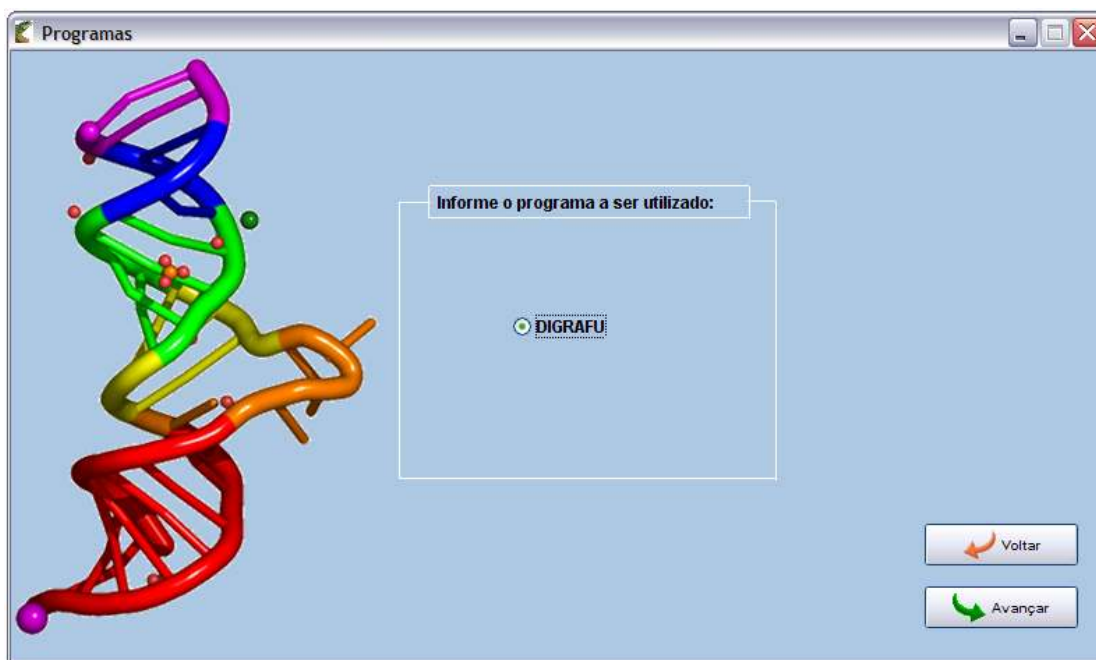


Figura 15 - Tela Programas – Distância.

Descrição: Tela onde o usuário escolhe qual programa de Distância deseja utilizar.

2.2.10 TELA_PHYML_SEQÜÊNCIA

Campo	Ação
Botão “Executar”	Abre a caixa de diálogo “Tipo de Execução”.
Botão “Voltar”	Abre a tela anterior “Programas”.
Botão “Editor”	Abre a tela “Editor”.
Botão “Visualizar”	Abre o “Hypertree”.

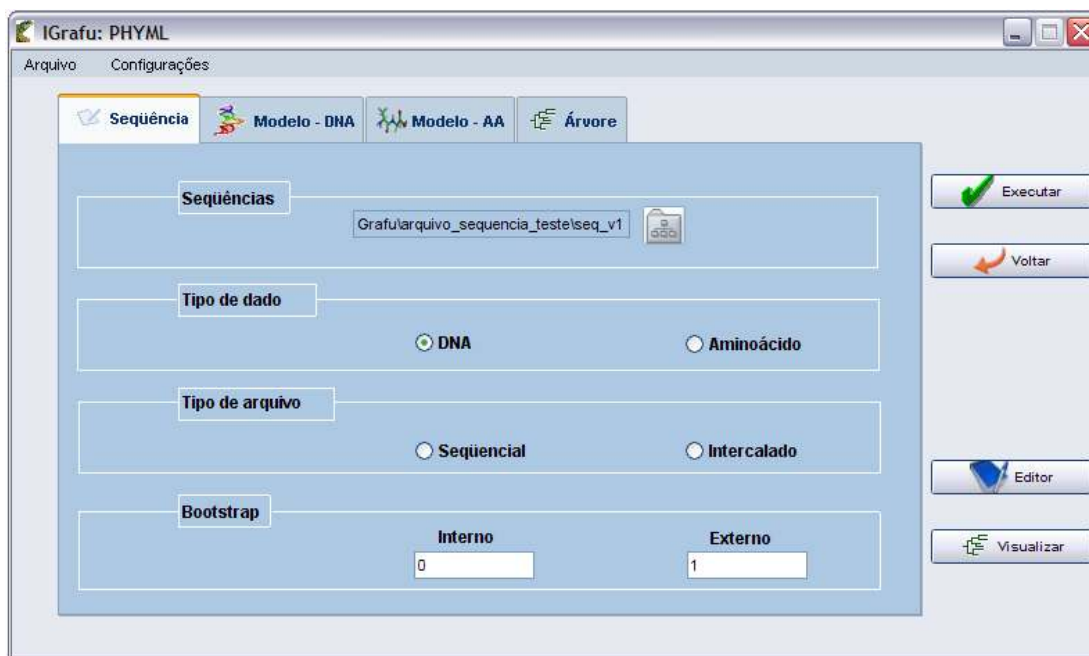


Figura 16 - Tela phym1 – Seqüência.

Descrição: Tela onde o usuário informa todos os dados referentes ao arquivo da seqüência passado para o phym1.

2.2.11 TELA_PHYML_MODELO_DNA

Campo	Ação
Botão “Executar”	Abre a caixa de diálogo “Tipo de Execução”.
Botão “Voltar”	Abre a tela anterior “Programas”.
Botão “Editor”	Abre a tela “Editor”.
Botão “Visualizar”	Abre o “Hypertree”.

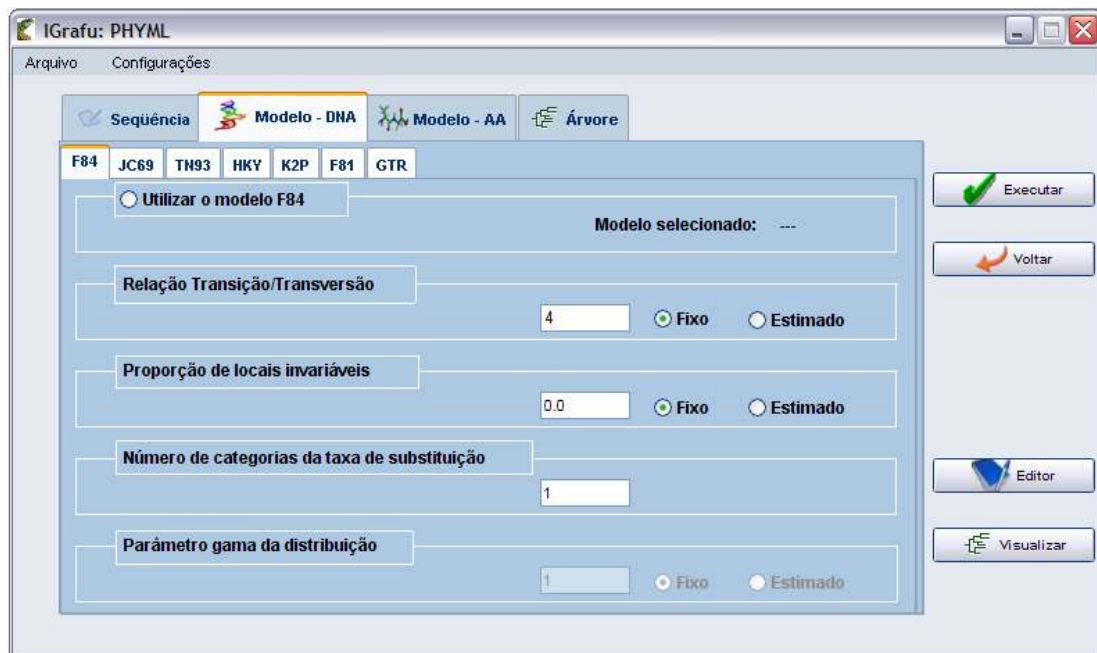


Figura 17 - Tela phymI – Modelo – DNA.

Descrição: Tela onde o usuário informa qual modelo de DNA deseja utilizar e seus respectivos parâmetros.

2.2.12 TELA_PHYML_MODELO_AA

Campo	Ação
Botão “Executar”	Abre a caixa de diálogo “Tipo de Execução”.
Botão “Voltar”	Abre a tela anterior “Programas”.
Botão “Editor”	Abre a tela “Editor”.
Botão “Visualizar”	Abre o “Hypertree”.

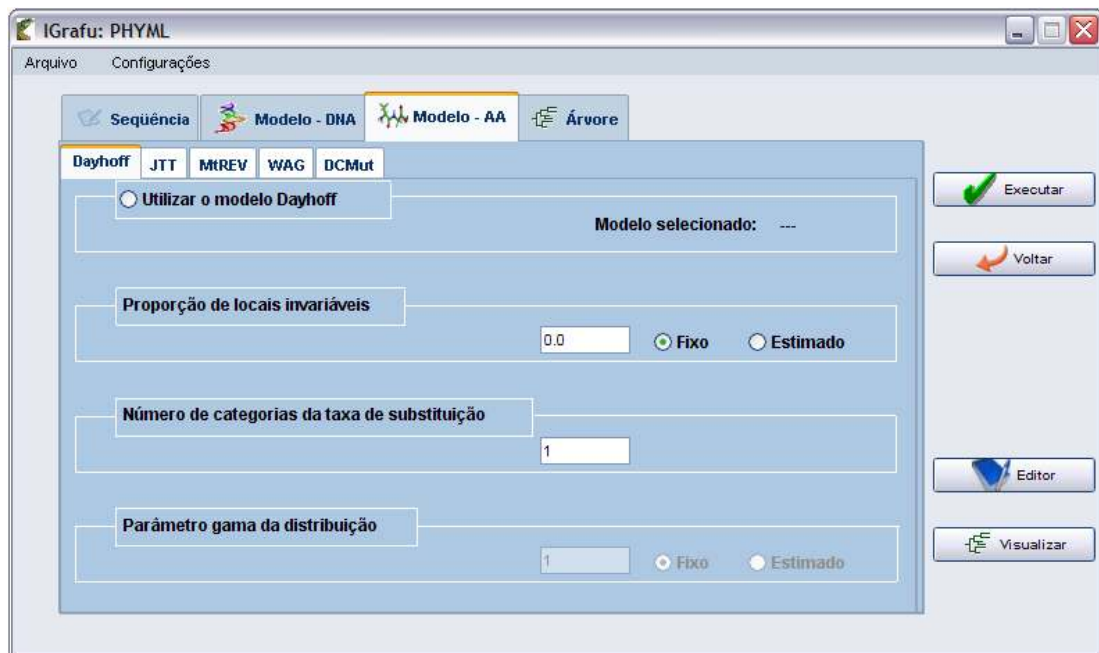


Figura 18 - Tela phym1 – Modelo – AA.

Descrição: Tela onde o usuário informa qual modelo de AA deseja utilizar e seus respectivos parâmetros.

2.2.13 TELA_PHYML_ÁRVORE

Campo	Ação
Botão “Executar”	Abre a caixa de diálogo “Tipo de Execução”.
Botão “Voltar”	Abre a tela anterior “Programas”.
Botão “Editor”	Abre a tela “Editor”.
Botão “Visualizar”	Abre o “Hypertree”.

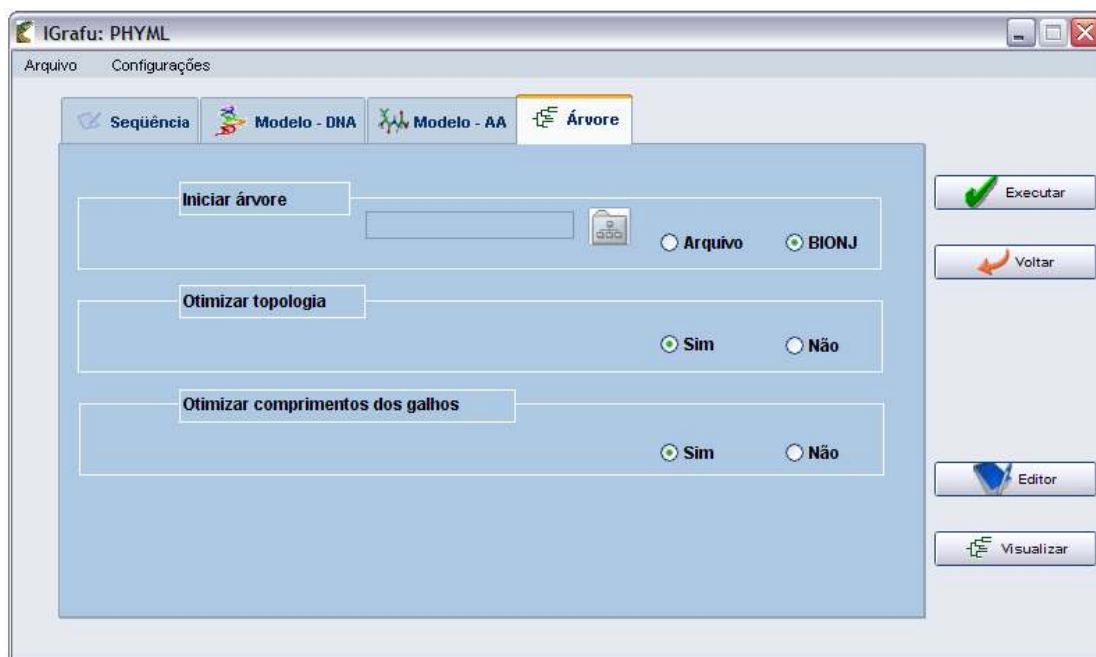


Figura 19 - Tela phymI – Árvore.

Descrição: Tela onde o usuário informa os dados referente à árvore filogenética a ser gerada pelo phymI.

2.2.14 TELA_PHYML_TIPO_EXECUÇÃO

Campo	Ação
Botão "Executar"	Executa o phymI.

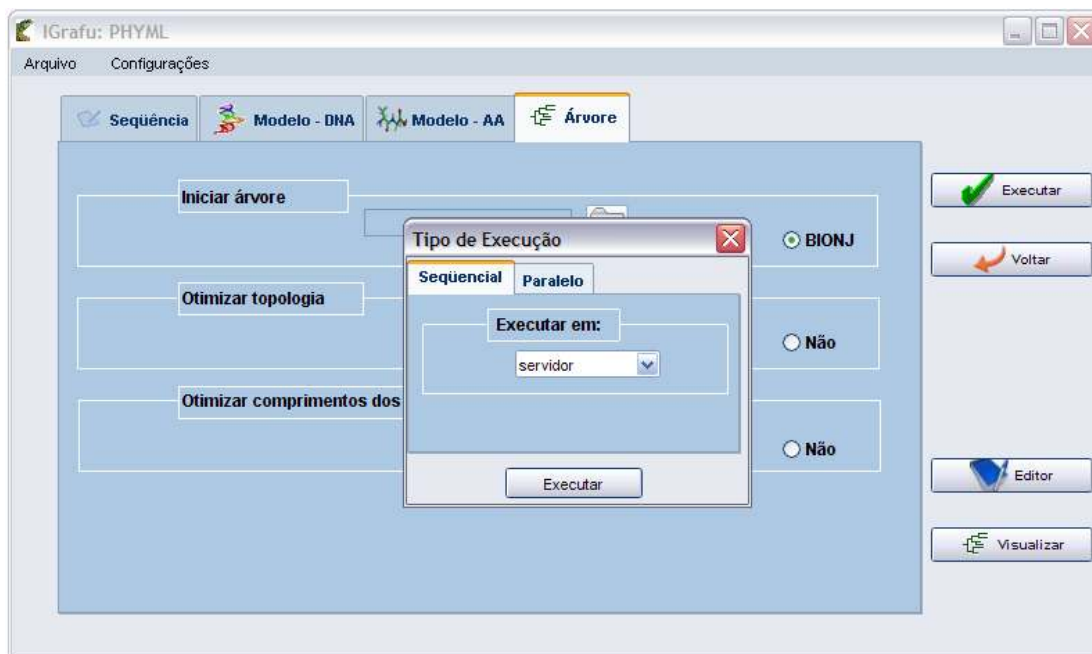


Figura 20 - Tela phymI Tipo de Execução.

Descrição: Tela onde o usuário informa o tipo de execução: Sequencial ou Paralelo. Caso sequencial, informar em qual nó do cluster deseja executar. Caso Paralelo, informar a quantidade de processadores que deseja utilizar.

2.2.15 TELA_DIGRAFU_SEQÜÊNCIA

Campo	Ação
Botão “Executar”	Abre a caixa de diálogo “Local de Execução”.
Botão “Voltar”	Abre a tela anterior “Programas”.
Botão “Editor”	Abre a tela “Editor”.
Botão “Visualizar”	Abre o “Hypertree”.

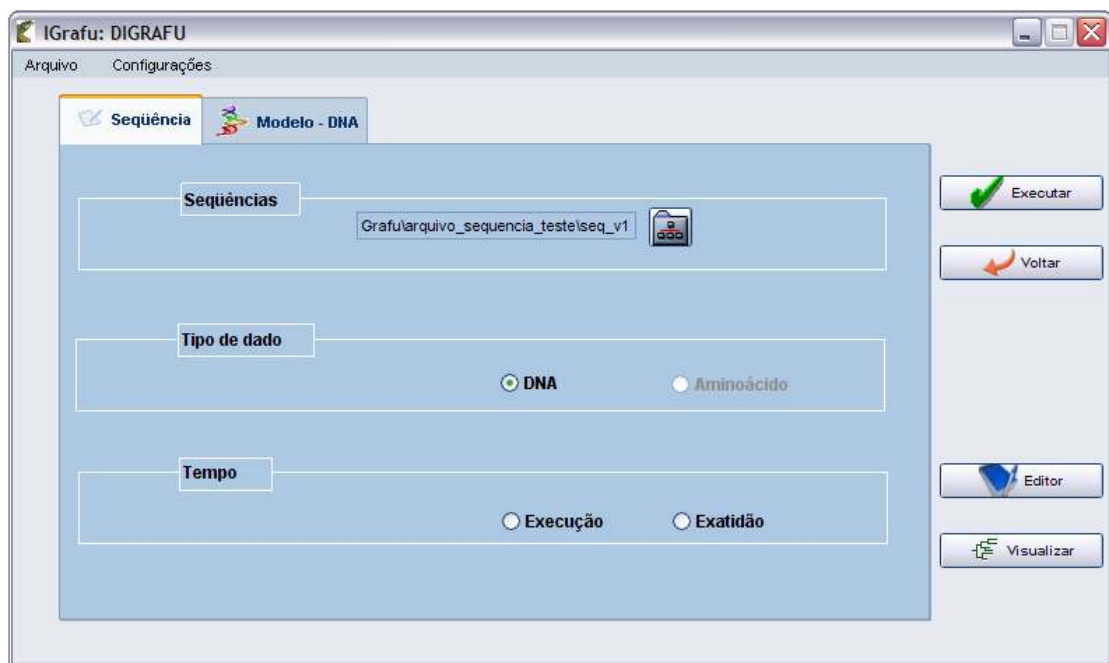


Figura 21 - Tela DiGrafu Seqüência.

Descrição: onde o usuário informa todos os dados referente ao arquivo da seqüência passado para o DiGrafu.

2.2.16 TELA_DIGRAFU_MODELO_DNA

Campo	Ação
Botão “Executar”	Abre a caixa de diálogo “Local de Execução”.
Botão “Voltar”	Abre a tela anterior “Programas”.
Botão “Editor”	Abre a tela “Editor”.
Botão “Visualizar”	Abre o “Hypertree”.

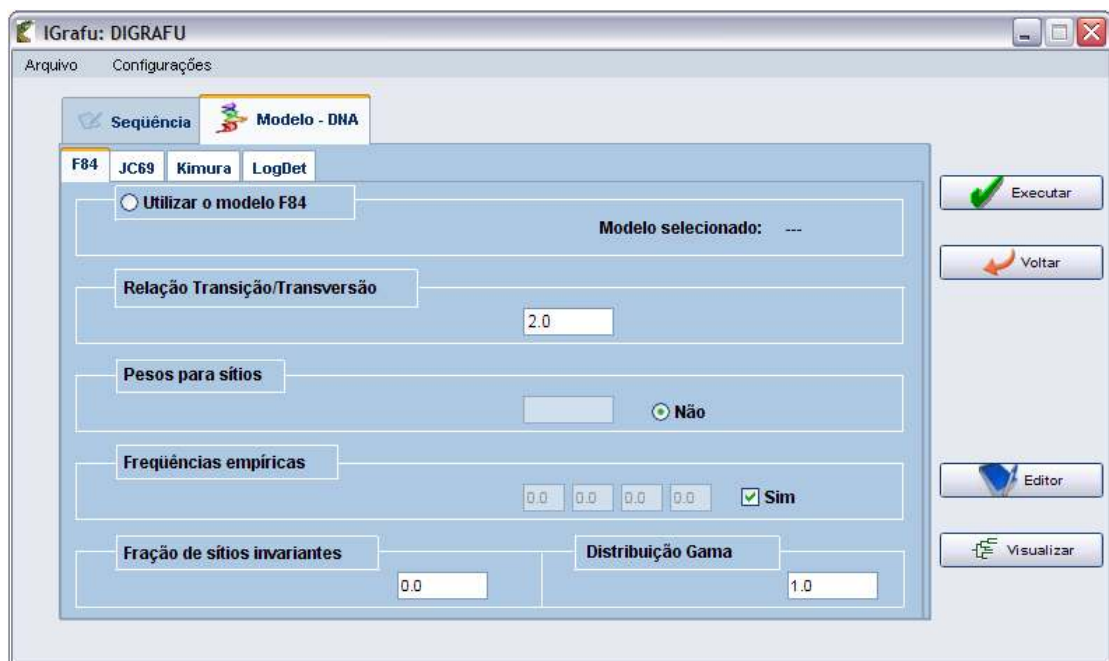


Figura 22 - Tela DiGrafu – Modelo – DNA.

Descrição: Tela onde o usuário informa qual modelo de DNA deseja utilizar e seus respectivos parâmetros.

2.2.17 TELA_DIGRAFU_LOCAL_EXECUÇÃO

Campo	Ação
Botão “Executar”	Executa o DiGrafu.

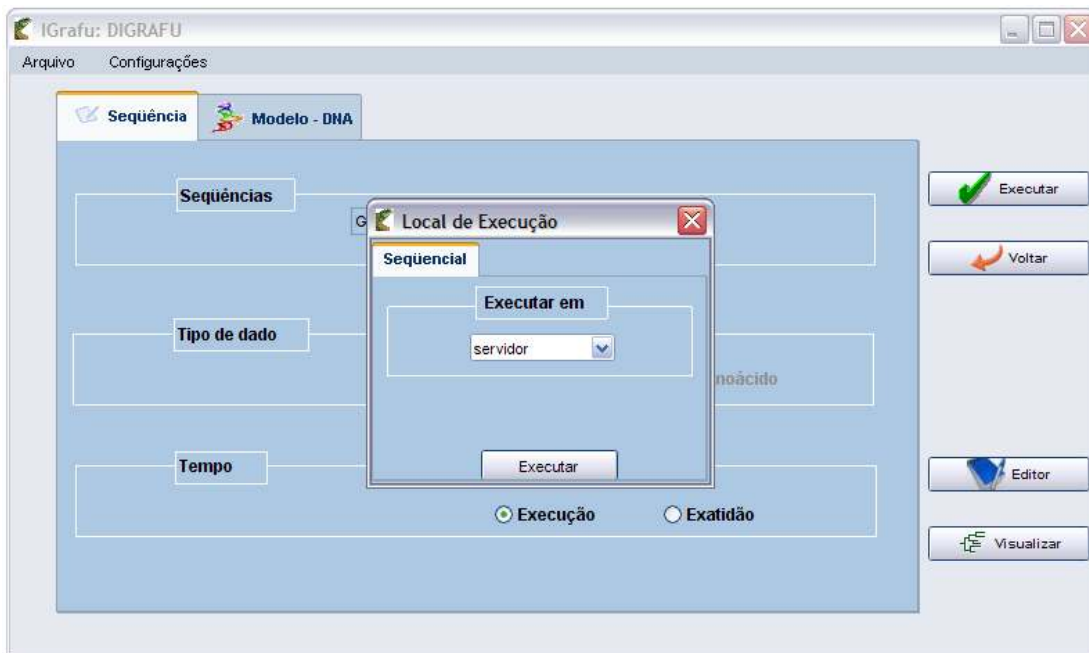


Figura 23 - Tela DiGrafu Local de Execução.

Descrição: onde o usuário informa o local de execução do DiGrafu, em qual nó do cluster deseja executar.

2.2.18 TELA_EDITOR

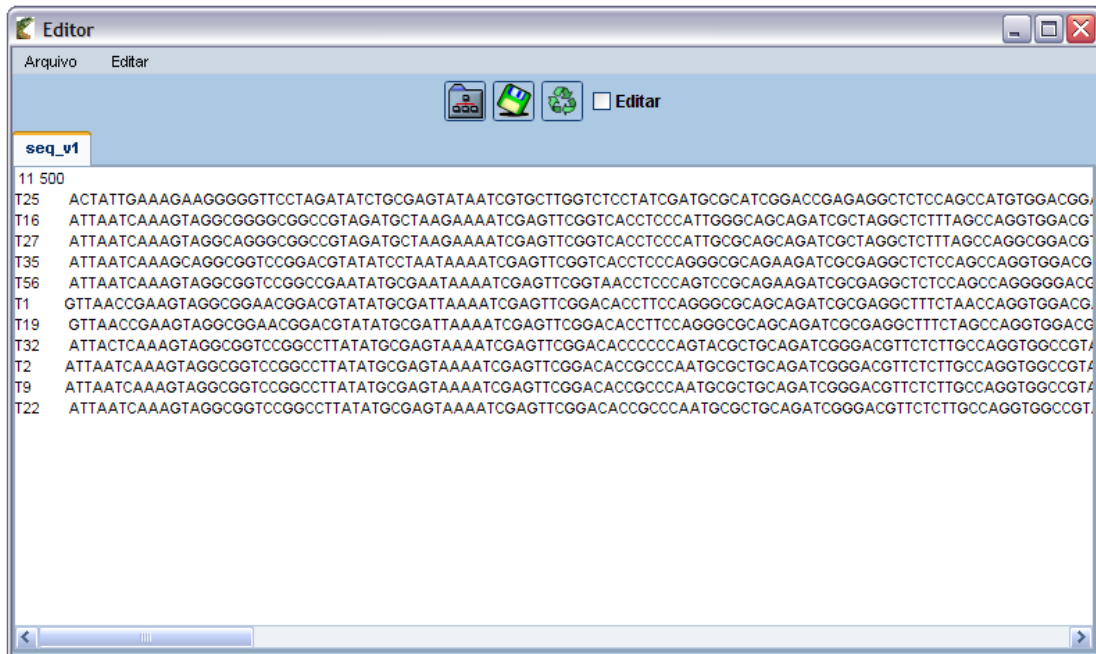


Figura 24 - Tela Editor.

Descrição: onde o usuário cria, edita ou gera texto (.txt) de entrada ou saída utilizados na execução da IGrafu.

2.2.19 TELA_HYPERTREE

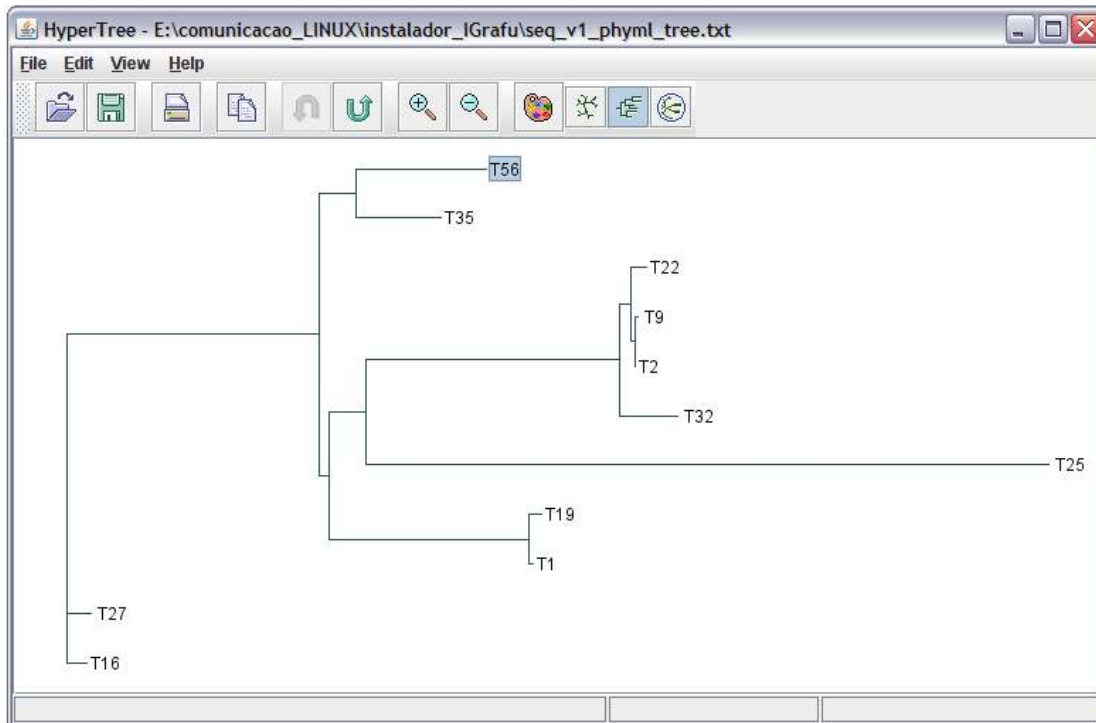


Figura 25 - Tela Hypertree.

Descrição: Tela onde o usuário visualiza a árvore filogenética gerada após a execução dos programas.