WELLISSON RODRIGO SANTOS GONÇALVES

PDBEST - UMA FERRAMENTA DE PROCESSAMENTO PARALELO PARA AQUISIÇÃO, MANIPULAÇÃO, EDIÇÃO E FILTRAGEM DE ARQUIVOS DO PROTEIN DATA BANK (PDB)

Belo Horizonte
29 de junho de 2015

Universidade Federal de Minas Gerais
Instituto de Ciências Biológicas
Programa de Pós-Graduação em Bioinformática

PDBEST - UMA FERRAMENTA DE PROCESSAMENTO PARALELO PARA AQUISIÇÃO, MANIPULAÇÃO, EDIÇÃO E FILTRAGEM DE ARQUIVOS DO PROTEIN DATA BANK (PDB)

Dissertação apresentada ao Curso de Pós-Graduação em Bioinformática da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Bioinformática.

WELLISSON RODRIGO SANTOS GONÇALVES

Belo Horizonte
29 de junho de 2015

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PDBest - Uma ferramenta de processamento paralelo para aquisição, manipulação, edição e filtragem de arquivos do Protein Data Bank (PDB)

WELLISSON RODRIGO SANTOS GONÇALVES

Ph. D. Raquel Cardoso de Melo-Minardi – Orientadora
Universidade Federal de Minas Gerais

Dra. Valdete Maria Gonçalves de Almeida – Co-orientadora
Universidade Federal de Minas Gerais

Belo Horizonte, 29 de junho de 2015

Resumo

Certamente, o Protein Data Bank (PDB), é o principal repositório de dados estruturais de biomoléculas e nas últimas décadas vem crescendo exponencialmente. Essa base de dados contém estruturas de biomoléculas tais como proteínas, DNAs, RNAs e seus ligantes. Seus dados são organizados em arquivos textuais nos quais, além das coordenadas atômicas, uma série de outras informações adicionais são apresentadas.

Por motivos diversos, frequentemente, observamos erros ou omissões nestes dados, o que torna árdua, a tarefa de manipulação de um grande volume de dados, que precisam ser tratados previamente antes de serem utilizados em análises através da Bioinformática.

Neste aspecto, há uma constante procura por ferramentas computacionais que tenham capacidade de tratar, ordenar, melhorar e uniformizar os arquivos PDB. Na tentativa de responder a estas necessidades, o PDBest foi criado. Este software pode acessar dados locais e consultar diretamente o RCSB[1], possibilitando de forma simples e direta os mais elaborados filtros disponíveis neste site para consulta.

Dentre suas principais funções, é possível separar um arquivo em suas respectivas cadeias; preservar ou retirar atributos de cabeçalho tais como HEADER, TITLE e REMARKS; remover, conservar ou indicar a quantidade de cada estrutura secundária no arquivo, selecionando HELIX e/ou SHEET além de selecionar apenas átomos da cadeia principal ou alguns resíduos específicos, apenas para citar alguns exemplos de uso.

Um outro ponto bastante importante é a possibilidade de localizar inconsistências nos arquivos identificando de que tipo são, o que é uma importante contribuição no sentido de melhorar e garantir a qualidade da base de dados a ser trabalhada. Há inúmeras inconsistências representativas nas estruturas resolvidas que podem comprometer a análise dos dados, como por exemplo a falta de átomos, a falta de resíduos e a presença de átomos com múltiplas ocupâncias.

E, por fim, para facilitar o compartilhamento de informações e a reprodutibilidade de experimentos em Bioinformática, o PDBest possui uma funcionalidade que extrai do programa, um arquivo de protocolo contendo um determinado fluxo de trabalho, incluindo as buscas na base de dados e o tratamento que foi dado aos arquivos, possibilitando anexar este protocolo a um estudo ou usar de modelo para fins de ensino.

[1] http://www.rcsb.org

A Dúvida...
Todos os tempos são de dúvida.
Mas de dúvida parecem ser, em particular,
As épocas conturbadas em que desaparecem todos os referenciais dos sentidos.

Dedico este trabalho a todos que possam
um dia se interessar por ele.
Estes são a razão de todo meu esforço.

Agradecimentos

Registro meus agradecimentos, a todos os que compartilharam o trilhar de mais este caminho, contribuindo direta ou indiretamente para a realização desta pesquisa, auxiliando-me e dando-me forças nos momentos em que precisei. Embora não possa citar a todos nominalmente. Alguns, não teria como deixar de citar, de modo algum.

Inicio agradecendo minha família pela presença constante em minha vida, na certeza que este trabalho se trata de uma pequena contribuição à ciência, mas um grande passo para minha formação.

A Maria de Fátima Santos Gonçalves, minha amada mãe, por todos os momentos dedicados a mim, pelas palavras, pelos conselhos, pelo amor, pela honestidade, pelo afeto e pela amizade, sem as quais estou convicto, não chegaria até aqui.

A Lucélia Viviane de Souza, minha companheira, a quem devo muitas horas de ausência, e as quais nunca poderei recompensar, senão retribuindo todo carinho e compreensão dedicados a mim.

Aos companheiros do Laboratório de Bioinformática e Sistemas - LBS, que em meio a imersão em seus trabalhos, nunca deixaram de me responder uma dúvida sequer. Em especial a ajuda prestada pelos colegas: Douglas Eduardo Valente Pires, Laerte Mateus Rodrigues e Sandro Carvalho Izidoro, pela presteza dos auxílios em meus afazeres.

Ás minhas orientadoras, Raquel Cardoso de Melo Minardi e Valdete Maria Gonçalves de Almeida, as quais ouso me referir como amigas, que por muitas vezes extrapolaram suas obrigações acadêmicas, pela união do nosso grupo de pesquisa, e para fazerem parte do meu trabalho.

A todos os professores e funcionários do curso de pos-graduação em Bioinformática, que sempre atenderam os alunos deste curso com paciência, dedicação e eficiência.

Aos meus companheiros de estudos, espalhados pela Universidade Federal de Minas Gerais, vínculo que adquiri durante este trabalho, mas que permanecerá por toda a vida.

A Universidade Federal de Minas Gerais pela estrutura, sem a qual o meu trabalho não aconteceria, pela CAPES, por manter a iniciativa de investir em pesquisa, financiando projetos e acreditando no potencial humano, principalmente o brasileiro.

Meus sinceros agradecimentos a todos que acreditaram em minha caminhada.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O significado da palavra Bioinformática remete ao desenvolvimento e emprego de modelos e/ou técnicas computacionais e matemáticos com intuito de gerenciar e/ou analisar dados biológicos (Attwood T.K. e E., 2011). Com o avanço da tecnologia e o surgimento de demandas crescentes por geração de informação a partir dessa massa de dados sendo produzida, a Bioinformática assumiu um sentido mais amplo. Atualmente ela é uma ciência interdisciplinar que combina conhecimentos das disciplinas de Química, Física, Biologia, Ciência da Computação, Matemática e Estatística, para processar e analisar dados biológicos com o objetivo de descobrir conhecimento. Verli, 2014 divide a Bioinformática em dois pontos de vista: um deles, trata de uma abordagem mais tradicional, considerando problemas ligados às sequências de nucleotídeos e aminoácidos, enquanto a outra, mais relacionada a este estudo, denomina-se Bioinformática Estrutural e se ocupa de questões biológicas relacionadas às propriedades tridimensionais, das estruturas moleculares, mais especificamente, das proteínas.

Considerando o tratamento de problemas, relacionados aos arquivos contendo dados sobre as estruturas de proteínas, alguns trabalhos têm se consolidado durante o tempo neste sentido. Os principais esforços que se evidenciam para manter a integridade dos dados, de forma a garantir as características especificadas no formato dos arquivos, são as dos próprios mantenedores dos arquivos (Berman et al., 2000, 2003; Boutselakis et al., 2003; Berman et al., 2007). Entretanto, é comum encontrar átomos e/ou resíduos ausentes e átomos com múltiplas ocupâncias, limitações intrínsecas às técnicas de obtenção da estrutura.

Paralelamente, colaboradores tem se empenhado a evoluir as linguagens de programação e em desenvolver bibliotecas (classes e métodos) (Stajich et al., 2002; Cock et al., 2009) que lidem com tarefas comuns ao tratamento de dados biológicos. Esta abordagem é muito poderosa e bem sucedida, entretanto há um alto custo atrelado ao poder que esta abordagem oferece, uma vez que é preciso dominar a lógica da programação bem como as linguagens a serem utilizadas. Nesse sentido, muitas pessoas preferem usar tecnologias que apresentem uma curva de aprendizado menor. Alunos de graduação ou outros pes-

quisadores mais iniciantes geralmente estão focados em sua área de formação e em alguns casos, podem achar inconveniente a complexidade para processar e padronizar, os dados antes do emprego de outras técnicas de análise.

Membros do Laboratório de Bioinformática e Sistemas[1] da UFMG, veem tratando omissões e inconsistências em seus arquivos PDB, desenvolvendo soluções para lidar com esta base de dados desde a formação do grupo em 2003. Pires et al. (2007) noticiou o embrião que originou o presente trabalho. Em seu estudo, foram desenvolvidos algoritmos que tratavam problemas como átomos e resíduos ausentes, filtragem por modelos e múltiplas ocupâncias. Desde então, os integrantes do grupo, passaram a usar, com bastante frequência, esses algoritmos, disponibilizados em scripts na linguagem perl e percebemos a necessidade de se investir em projetar e implementar uma ferramenta que pudesse ser amplamente utilizada por pesquisadores que necessitem trabalhar com dados de estruturas de proteínas.

1.1 Objetivos

Nessa seção, descrevemos o objetivo geral e os específicos que guiaram o desenvolvimento dessa dissertação.

1.1.1 Objetivo geral

Projetar e implementar uma ferramenta para coletar, filtrar (segundo critérios específicos), converter, identificar omissões e melhorar a qualidade de arquivos com coordenadas estruturais de biomoléculas em conformidade com o consórcio wwPDB (Berman et al., 2007).

1.1.2 Objetivos específicos

- Projetar e implementar um software capaz de identificar e manipular um conjunto de arquivos PDB objetivando aumentar sua qualidade, mantendo tempo de processamento aceitável.
- Projetar e implementar um site para comunicação com a comunidade científica e disponibilização do software e de sua documentação.
- Desenvolver um manual para o software implementado, para facilitar o entendimento dos utilizadores.
- Testar e avaliar a utilidade da ferramenta através de estudos de casos.
- Produção de um artigo científico descrevendo a ferramenta desenvolvida.

[1] http://www.lbs.dcc.ufmg.br

1.1.3 Organização

Este trabalho está organizado da seguinte forma:

- Uma pequena introdução, já apresentada, contendo as principais motivações para implementação do software.
- A construção de um referencial teórico como base deste estudo.
- As metodologias adotadas para a construção do software.
- E por último, os resultados, as conclusões e os trabalhos futuros que surgirão a partir deste estudo.

Capítulo 2

Fundamentação Teórica

2.1 Bioinformática estrutural

Devido a alta disponibilidade de dados, provocada pelo crescimento e abrangência dos projetos de pesquisa em Ciências Biológicas ao redor do mundo, são grandes as demandas por armazenamento, gerência e análise de crescentes volumes de dados biológicos.Em tempos pós-genômicos, a proteômica, aliada ao entendimento funcional e estrutural das proteínas, tem ganhado cada vez mais representatividade (da Silveira, 2005; Dorn e Norberto de Souza, 2013).

A crescente evolução dos processos experimentais, como a Cristalografia por difração de raios X e a Ressonância magnética nuclear (Berg et al., 2004), juntamente com técnicas computacionais para previsão da estrutura tridimensional ou a forma com que ocorre o enovelamento de uma certa sequência de aminoácidos, como modelagem comparativa por homologia e métodos *ab initio* (Dorn e Norberto de Souza, 2013), tem provocado o aumento no volume de informações estruturais, estimulando a criação de uma subárea da Bioinformática, denominada Bioinformática Estrutural.

A Bioinformática Estrutural é o estudo das estruturas biomoleculares e suas propriedades físico-químicas, através da aplicação de técnicas de Bioinformática, contribuindo assim para explorar, organizar e compreender informações estruturais associadas a essas moléculas (Luscombe et al., 2001), tendo como foco principal a representação, armazenamento, recuperação, análise e visualização das informações estruturais de proteínas e polipeptídeos (da Silveira, 2005; Dorn e Norberto de Souza, 2013).

É possível dizer que a função de uma proteína ou enzima está fortemente relacionada a sua estrutura tridimensional e conforme Lesk, 2001 e Berg et al. (2004) sugeriram, as proteínas que pertencem a uma mesma família funcional, mesmo com uma baixa similaridade de sequência, tem estruturas tridimensionais conservadas. Um bom exemplo são as globinas, que apresentam sequências muito pouco conservadas, mas estruturas tridimensionais muito similares. A proteômica utiliza componentes estruturais para entender o enovelamento proteico, processo pelo qual os resíduos de uma sequência se dobram de

forma orientada, buscando posições tais, que formem uma proteína nativa funcional (Lesk, 2001).

2.2 Aminoácidos

Isoladamente, os aminoácidos possuem uma organização básica semelhante: um átomo de Carbono, localizado ao centro da estrutura, denominado de Carbono α (Cα) ligado a quatro grupos de ligantes: um átomo de hidrogênio (H), um grupamento amino (-NH_2), um grupamento carboxílico (-COOH) e um grupamento orgânico R, também chamado de cadeia lateral como é possível observar na Figura 2.1 (Voet et al., 2014).

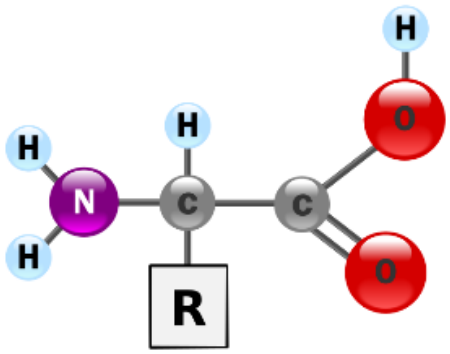


Figura 2.1: Estrutura geral de um aminoácido.

Fonte: http://www.explicatorium.com/quimica/Proteinas.php

Dois aminoácidos se ligam quando um grupo carboxílico (-COOH), de uma molécula de um certo aminoácido, reage com o grupo amina (-NH_2) de outro, liberando uma molécula de água (H_2O), como na Figura 2.2. Desta ligação covalente entre o átomo de carbono (C) e o nitrogênio (N), a chamada ligação peptídica, resulta em um dipeptídeo que quando ligados a outros destes, formam os polipeptídios dando origem as proteínas.

Por convenções internacionais, os aminoácidos são identificados por abreviações de três letras (derivadas dos nomes em inglês), ou por um código de uma letra (Nelson e Cox, 2011; Berg et al., 2004) como mostra a Tabela 2.1. Conforme Nelson e Cox (2011), a natureza química das cadeias laterais (grupos R), classifica os aminoácidos em grupos: Apolares, Polares, Polares Básicos e Polares Ácidos e o entendimento destas propriedades físico-químicas nos aminoácidos é importante a medida em que, são responsáveis pela estabilidade e funcionamento da proteína.

2.3 Proteínas

As proteínas são macromoléculas biológicas, compostas por resíduos de aminoácidos, unidos por ligações peptídicas. A Figura 2.3 mostra o alfabeto dos 20 aminoácidos comumente encontrados nos seres vivos.

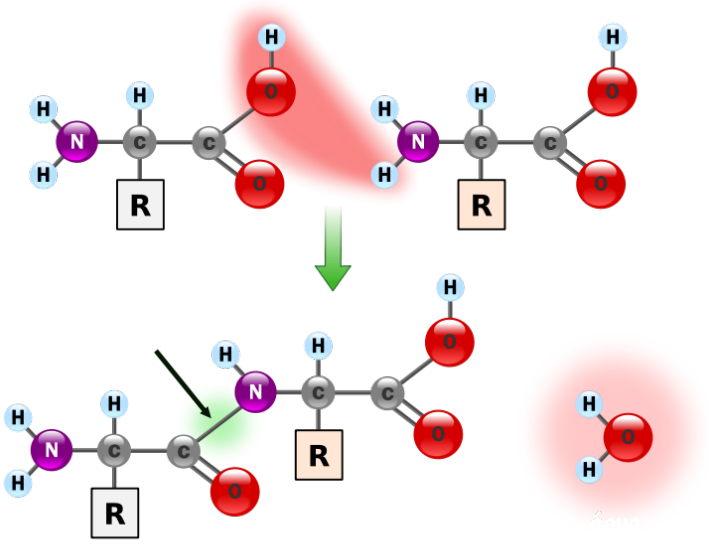


Figura 2.2: Formação da ligação peptídica

Fonte: http://www.explicatorium.com/quimica/Proteinas.php

Tabela 2.1: Nomes, símbolos e abreviações dos aminoácidos

Nome	Símbolo	Abreviação	Nome	Símbolo	Abreviação
Glicina ou Glicocola	Gly, Gli	G	Tirosina	Tyr, Tir	Y
Alanina	Ala	A	Asparagina	Asn	N
Leucina	Leu	L	Glutamina	Gln	Q
Valina	Val	V	Isoleucina	Ile	I
Glutamato ou Ácido glutâmico	Glu	E	Aspartato ou Ácido aspártico	Asp	D
Prolina	Pro	P	Arginina	Arg	R
Fenilalanina	Phe ou Fen	F	Lisina	Lys, Lis	K
Serina	Ser	S	Histidina	His	H
Treonina	Thr, The	T	Triptofano	Trp, Tri	W
Cisteina	Cys, Cis	C	Metionina	Met	M

Fonte: Nelson e Cox (2011)

A sequência de resíduos é formada através de ligações peptídicas e posteriormente esta sequência primária adota uma conformação tridimensional única e é esta estrutura que determina a função da proteína na célula (C. e J., 1999; Andreas D. Baxevanis, 2001). O estudo destas conformações possibilita construir conhecimento a cerca de suas funções e os aspectos gerais do enovelamento protéico (Lesk, 2014).

2.4 A Base de dados PDB (*Protein Data Bank*)

A base de dados PDB foi estabelecida em 1971, pelo *Brookhaven National Laboratories (BNL)*, como um local para depósito de arquivos para descrever estruturas biológicas macromoleculares (Bernstein et al., 1977; Berman et al., 2000). Com sete estruturas depositadas em meados de 1980, teve uma ascensão considerável, no volume de estrutu-

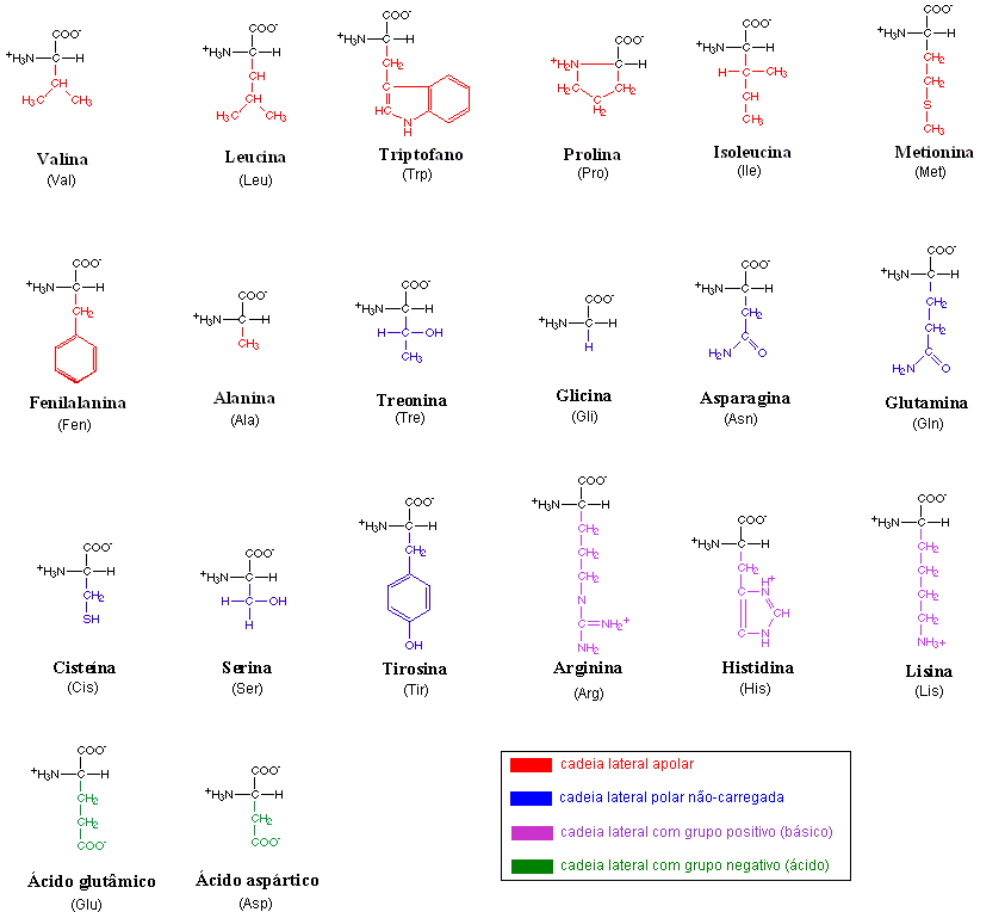


Figura 2.3: Aminoácidos que compõem as proteínas

ras depositadas, devido a avanços das tecnologias, em todos os aspectos que envolvem a obtenção estrutural e nas mudanças na visão da comunidade científica sobre o compartilhamento de dados. Essas melhorias refletiram no principal processo de obtenção da estrutura, a Cristalografia, e com a utilização de um novo modelo na época, o processo de Ressonância Magnética Nuclear (NMR). Atualmente, o PDB conta com mais de 109.274 estruturas[1], obtidas através de diversos métodos, sendo que o mais representativo é o método de Cristalografia por Difração de Raios X, como mostrado na Tabela 2.2.

Tabela 2.2: Percentual de tipos experimentos para resolução de estruturas dos arquivos depositados no PDB

Exp.Method	Proteins	Nucleic Acids	Protein/NA Complexes	Other	Total	%
X-RAY	89.167	1.610	4.402	4	95.183	89,07%
NMR	9.515	1.112	224	8	10.859	10,16%
ELECTRON MICROSCOPY	542	29	175	0	746	0,70%
HYBRID	68	3	2	1	74	0,07%

Fonte: http://www.rcsb.org/pdb/statistics/holdings.do

[1] http://www.rcsb.org

2.4.1 Depósito, validação e disponibilização das estruturas

Para que o arquivo estrutural seja disponibilizado na base de dados PDB, Berman et al. (2000) cita a necessidade de três passos importantes: o depósito dos dados, a anotação e a validação, que são parte do sistema de processamento de dados no processo de inserção de uma nova estrutura, como apresentado na Figura 2.4.

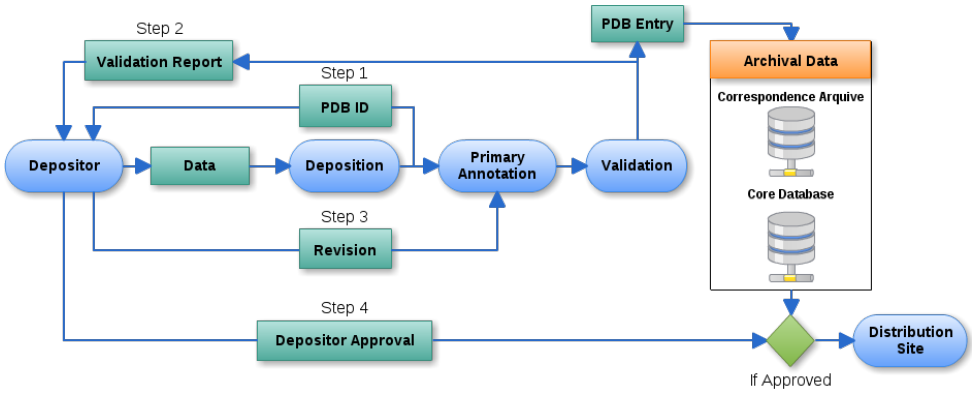


Figura 2.4: Passos para depósito de uma estrutura no PDB

Fonte: Berman et al. (2000)

A Figura 2.4 mostra um conjunto de ações, para o gerenciamento do processo de depósito da estrutura, adotado quando da criação do banco de dados. Ao iniciar o processo, o autor recebe um identificador e inicia o processo de validação, para diagnosticar erros e inconsistências nos dados (*Step 1*). Após o processo de validação, uma estrutura contendo os dados gerados, pelo processo de validação, é remetido de volta ao depositante (*Step 2*). Depois de analisar o arquivo processado, o autor envia as revisões que achar necessárias (*Step 3*). Dependendo da natureza das revisões, os passos anteriores podem ser repetidos ciclicamente. Concluído esta fase e recebida a aprovação pelo autor, no passo seguinte (*Step 4*), a base de dados está pronta para distribuir a nova estrutura (Berman et al., 2000).

A validação consiste, na avaliação da qualidade dos modelos atômicos depositados. Ao depositar sua estrutura, o responsável pelo depósito, recebe uma carta informando que as seguintes validações serão executadas:

- **Distâncias de ângulos das ligações covalentes:** As proteínas são comparadas com valores padrões de Engh e Huber (1991); bases de ácidos nucleicos são comparados com os valores normais de Gelbin et al. (1996); açúcares e fosfatos são comparados com valores normais de Gelbin et al. (1996).

- **Validação estereoquímica:** Todos os centros quirais de proteínas e ácidos nucleicos são verificados quanto à estereoquímica correta (Westbrook et al., 2003).

- **Nomenclatura atômica:** Verificação e ajuste da nomenclatura de todos os átomos de acordo com as normas da Rich et al. (1983).

- **Contatos próximos:** As distâncias entre todos os átomos dentro da unidade assimétrica de estruturas cristalinas e a molécula única de estruturas de NMR são calculados. Para as estruturas de cristal, os contatos entre as moléculas relacionadas com a simetria também são verificados Westbrook et al. (2003).

- **Nomenclatura de ligantes e átomos:** Nomenclatura de resíduos e átomos são comparados com o dicionário PDB para todos os ligantes, bem como de resíduos e bases normalizadas. Grupos de ligantes não reconhecidos são sinalizados e qualquer discrepância nos ligantes conhecidos são listados como átomos extras ou faltantes (Westbrook et al., 2003).

- **Comparação de sequências:** A sequência de dados nos registros PDB SEQRES é comparada com a sequência gerada a partir dos registos das coordenadas. Após esta comparação as informações são processadas de forma que as eventuais diferenças ou resíduos ausentes sejam selecionados. Durante o processamento da estrutura as referências do banco de dados de sequência, dadas por DBREF e SEQADV, são verificados quanto à precisão. Se nenhuma referência é fornecida, o BLAST é usado para encontrar a melhor combinação (Berman et al., 2000). Qualquer conflito entre os registros PDB SEQRES e as sequências derivadas dos registros das coordenadas é resolvido por comparação entre vários bancos de dados de sequência (Westbrook et al., 2003).

- **Águas distantes:** As distâncias entre todos os átomos de oxigênio, das moléculas de água e todos os átomos polares (oxigênio e nitrogênio), das macromoléculas, ligantes e solventes, são calculadas. Os átomos do solvente que estejam distantes são reposicionados, usando um processo de simetria cristalográfica, de tal forma a serem inseridos na esfera de solvatação da estrutura (Westbrook et al., 2003).

Embora vários casos de erros graves sejam descobertos e solucionados nesta fase, e a proposta do PDB seja tornar o arquivo consistente, de tal maneira que, esteja completamente livre de erros (Berman et al., 2000; Westbrook et al., 2003), há ainda diversos tipos de incertezas, resultantes dos vários métodos para obtenção das estruturas que, muitas vezes, não conseguem prever com precisão a estrutura.

2.5 O Consórcio do banco de dados wwPDB

Criado em 2003, o consórcio wwPDB busca formalizar o livre acesso e a natureza internacional do conteúdo disponível nos repositórios do PDB (Berman et al., 2007). Esta organização é fruto de colaboração internacional para gerenciar os depósitos e distribuições dos arquivos estruturais de biomoléculas (Henrick K., 2005). Atualmente, com mais de 109.274 estruturas depositadas[2], incluindo proteínas, ácidos nucleicos e grandes complexos

[2]http://www.rcsb.org

macromoleculares, os quais foram determinados por diversos métodos experimentais, tais como Cristalografia por Difração de Raios X, Ressonância Magnética Nuclear (RNM) e Microscopia Eletrônica, o Consórcio conta com quatro espelhos repositórios (*mirrors*) espalhados pelo mundo: RCSB PDB (EUA), MSD-EBI (Europa), PDBj (Japão) e o grupo BMRB (EUA) se juntou ao wwPDB em 2006 (Berman et al., 2003).

Os fundadores desta iniciativa foram: RCSB PDB (EUA) (Berman et al., 2000), *Macromolecular Structure Database at the European Bioinformatics Institute* (MSD-EBI) (EUR) (Golovin et al., 2004) e o PDB *Japan* (PDBj) (JPN) da *Osaka University* (Berman et al., 2007). A Tabela 2.3 indica o acesso e fornece o caminho dos locais para depósito e a distribuição dos arquivos. O *BioMagResBank* (BMRB) da *University of Wisconsin-Madison* (USA) (Ulrich et al., 1988), tornou-se membro do consórcio wwPDB em 2006 e trata se de um repositório de dados obtidos através da técnica de RNM em proteínas, peptídeos, ácidos nucléicos, e outras biomoléculas.

Tabela 2.3: Locais de depósito e acesso a dados

	Access PDB FTP	Deposit data	Main website
RCSB PDB	ftp://ftp.rcsb.org/pub/pdb	http://deposit.rcsb.org	http://www.rcsb.org
MSD-EBI	ftp://ftp.ebi.ac.uk/pub/databases/rcsb/pdb	http://www.ebi.ac.uk/msd-srv/autodep4	http://www.ebi.ac.uk/msd
PDBj	ftp://pdb.protein.osaka-u.ac.jp/pub/pdb	http://www.pdbj.org/deposit.html	http://www.pdbj.org
BMRB	–não possue–	http://www.bmrb.wisc.edu/deposit/	http://www.bmrb.wisc.edu

Fonte: http://www.wwpdb.org

Cada espelho fornece acesso ao mesmo arquivo PDB, e tratativas particulares implementadas de maneira proprietária, com diferentes visões e análises dos dados estruturais depositados no PDB (Deshpande et al., 2005; Wako et al., 2004). A Figura 2.5 mostra a organização dos dados pelo mundo.

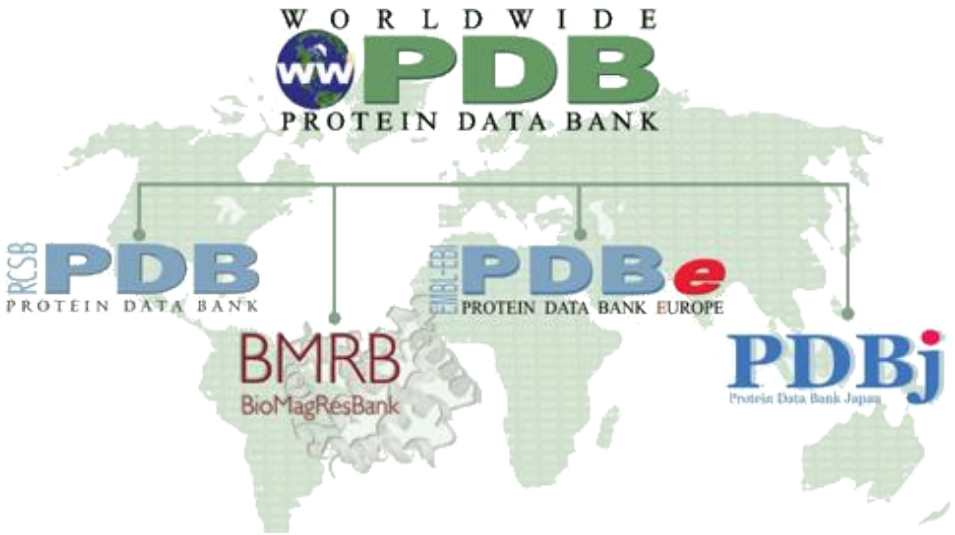


Figura 2.5: Mapeamento dos espelhos do PDB no mundo

Fonte: http://www.wwpdb.org

Embora o PDB se autodenomine como um banco de dados relacional, Schierz et al. (2007) sugere que não é bem assim, pois o que é disponibilizado para *download* são

arquivos contendo texto estruturado. Atualmente, é possível fazer o *download* destes arquivos em diversos formatos, e cada espelho fornece o conjunto de formatos, que julga ser o mais interessante. Nesse trabalho trataremos de arquivos no formato pdb, que serão descritos a seguir.

Trata-se de arquivos que descrevem estruturas tridimensionais de macromoléculas biológicas, determinadas experimentalmente. Ao longo do tempo, várias mudanças no formato dos arquivos foram realizadas e atualmente, na versão 3.3, os dados contidos nestes arquivos incluem coordenadas atômicas, informações de cristalografia de raios X ou dados de modelos, para estruturas obtidas por Ressonância Nuclear Magnética. Cada arquivo depositado também inclui os nomes das moléculas, informações da estrutura primária e secundária, referências ao banco de dados de sequência, de ligantes quando apropriado, detalhes sobre a obtenção dos dados, como a estrutura foi resolvida e citações.

Os arquivos PDB contêm uma série de linhas, cada qual é dividida em 80 colunas, sendo as seis primeiras colunas identificadoras do tipo do registro contido no restante da linha. Este conjunto de seis colunas é chamado de tipo de registro (*Record Type*) e cada registro pode ocupar uma, ou continuar por várias linhas. A Tabela 2.4 mostra a divisão destes registros em dez grandes grupos.

- **Sessão de Título:** Sessão usada para descrever os experimentos com as biomoléculas e comentários.
- **Sessão de Estrutura Primária:** Sessão usada para indicar sequência de resíduos em cada cadeia da estrutura.
- **Sessão de Átomos Heterogêneos:** Contém a relação completa dos átomos não convencionais.
- **Sessão de Estrutura Secundária:** Esta sessão trata da estrutura secundária da molécula, alfa hélices e folhas beta.
- **Sessão de Anotação de Conectividade:** Utilizado para anotações de conectividade, permite a especificação e localização de pontes dissulfeto.
- **Sessão de Funções Diversas:** Utilizado para descrição das diversas características da molécula ou de um sítio ativo.
- **Sessão de Transformação de Coordenadas e Dados Cristalográficos:** Esta sessão descreve a geometria da experiência de cristalografia e a transformações para os sistemas de coordenadas.
- **Sessão de Coordenadas:** Nesta sessão, está o conjunto de coordenadas atômicas e indicações de início e fim de modelo a qual o conjunto de coordenadas representa, no caso de obtenção de estrutura por NMR.

- **Sessão de Conectividade:** Esta sessão está relacionada com informações sobre a conectividade atômica.
- **Sessão de Registro:** Trata- se da indicação da finalização do arquivo.

Tabela 2.4: Guia de conteúdo para os itens do arquivo PDB

Item	Conteúdo		
Sessão de Título (*Title Section*)	Cabeçalho (*HEADER*)	Fonte (*SOURCE*)	Autor (*AUTHOR*)
	Obsoleto (*OBSLTE*)	Termos Relevantes (*KEYWDS*)	Histórico de Revisões (*REVDAT*)
	Título (*TITLE*)	Dados do experimento (*EXPDTA*)	Lista de Códigos obsoletos (*SPRSDE*)
	Entradas Relacionadas (*SPLT*)	Número de Modelos (*NUMMDL*)	Citação Primária na Literatura (*JRNL*)
	Alerta para Erros e Problemas não resolvidos na entrada (*CAVEAT*)	Anotação Adicional Relativo às Coordenadas (*MDLTYP*)	Observações (*REMARKS*)
	Conteúdo Molecular (*COMPND*)		
Sessão de Estrutura Primária (*Primary Structure Section*)	Referência Cruzada (*DBREF (standard format)*)	Diferença entre Informações de Sequências (*SEQADV*)	Registro de Modificações (*MODRES*)
	Extensão para o campo (*DBREF*) (*DBREF1/DBREF2*)		Lista de Resíduos (*SEQRES*)
Sessão de Resíduos Não Convencionais (*Heterogen Section*)	Resíduos Não Convencionais (*HET*)	Nome Químico do Composto (*HETNAM*)	Sinônimo do Composto (*HETSYN*)
	Fórmula química (*FORMUL*)		
Sessão da Estrutura Secundária (*Secondary Structure Section*)	Alpha Hélice (*HELIX*)	Folha Beta (*SHEET*)	
Sessão de Anotações de Conectividade (*Connectivity Annotation Section*)	Ligação Dissulfeto (*SSBOND*)	Conectividade não Implícita (*LINK*)	Identificação de Conformação CIS (*CISPEP*)
Sessão de Funções Diversas (*Miscellaneous Features Section*)	Co-Fator Catalítico (*SITE*)		
Sessão de Transformação de Coordenadas e Dados Cristalográficos (*Crystallographic and Coordinate Transformation Section*)	Parâmetros da Célula Unitária (*CRYST1*)	Parâmetros de Transformação de Entradas Ortogonais (*ORIGXn*)	Parâmetros de Transformação de Entradas Ortogonais das Coordenadas Cristalográficas (*SCALEn*)
	Registro de Transformação de Simetria Não-Cristalográfica (*MTRIXn*)		
Sessão de Coordenadas (*Coordinate Section*)	Número de Série do Modelo (*MODEL*)	Apresenta Fatores de Temperatura Anisotrópica (*ANISOU*)	Apresentação de Coordenadas Atômicas Para Resíduos ditos Não-Padrão (*HETATM*)
	Apresentação de Coordenadas Atômicas Para Resíduos ditos Padrão (*ATOM*)	Indica o Fim de uma Cadeia (*TER*)	Indica o Fim de um Modelo (*ENDMDL*)
Sessão de Conectividade (*Connectivity Section*)	Indica a Conectividade Entre os Átomos (*CONECT*)		
Sessão de Registro (*Bookkeeping Section*)	Lista Número de linhas na entrada de coordenadas. (*MASTER*)		Marca o Final de um arquivo PDB (*END*)

Fonte: http://www.wwpdb.org/documentation/file-format-content/format33/v3.3.html

A estrutura mostrada na Tabela 2.5 (*Crystal structure of Human Phosphodiesterase 5 complexed with Sildenafil*) contém coordenadas atômicas da proteína e seus ligantes, também é possível utilizar visualizadores gráficos para gerar uma imagem da estrutura 3D ou mesmo editores de texto. O primeiro bloco da tabela traz informações referentes à descrição da molécula, à classificação bioquímica, ao código único utilizado pelo PDB, ao título do trabalho publicado, o qual originou a estrutura. No segundo bloco, reservado aos átomos que formam a proteína e suas coordenadas atômicas, seguido do fator de ocupância, para todos os átomos. O terceiro bloco trata dos átomos dos ligantes, estes átomos não pertencem à estrutura da proteína, mas estão ligados a ela. O registro *HETATM* é atribuído a todos os átomos dos ligantes, inclusive metais e solventes, como por exemplo os íons, Zinco (ZN) e Magnésio (MG) que foram cristalizados durante o processo de cristalografia.

Tabela 2.5: Parte do arquivo PDB do complexo 1UDT

HEADER	HYDROLASE								06-MAY-03	1UDT
TITLE	CRYSTAL STRUCTURE OF HUMAN PHOSPHODIESTERASE 5 COMPLEXED									
TITLE	2 WITH SILDENAFIL(VIAGRA)									
COMPND	MOL_ID: 1;									
COMPND	MOLECULE: CGMP-SPECIFIC 3' ,5'-CYCLIC PHOSPHODIESTERASE									
ATOM	1	N	THR	A	537	-3.982	28.215	83.642	1.00	43.46	N
ATOM	2	CA	THR	A	537	-2.685	28.492	84.329	1.00	42.15	C
ATOM	3	C	THR	A	537	-2.853	28.484	85.842	1.00	42.35	C
ATOM	4	O	THR	A	537	-3.963	28.599	86.365	1.00	40.95	O
ATOM	5	CB	THR	A	537	-2.104	29.875	83.957	1.00	39.76	C
ATOM	6	OG1	THR	A	537	-2.892	30.899	84.577	1.00	34.38	O
ATOM	7	CG2	THR	A	537	-2.100	30.080	82.451	1.00	30.89	C
ATOM	8	N	ARG	A	538	-1.731	28.366	86.536	1.00	43.46	N
ATOM	9	CA	ARG	A	538	-1.722	28.343	87.987	1.00	44.97	C
ATOM	10	C	ARG	A	538	-2.256	29.661	88.522	1.00	43.58	C
HETATM	2544	ZN	ZN	A	1001	-4.452	56.769	81.496	1.00	33.15	ZN
HETATM	2545	MG	MG	A	1002	-3.709	57.206	77.758	1.00	27.93	MG
HETATM	2546	C34	VIA	A	1000	-0.132	61.467	80.120	1.00	52.65	C
HETATM	2547	C33	VIA	A	1000	-0.169	61.792	81.589	1.00	38.29	C
HETATM	2548	C32	VIA	A	1000	-1.261	62.864	81.927	1.00	38.55	C
HETATM	2549	C30	VIA	A	1000	-1.114	63.031	83.405	1.00	36.46	C
HETATM	2550	N29	VIA	A	1000	-1.798	62.280	84.370	1.00	36.25	N

Fonte: http://www.rcsb.org

As estruturas depositadas no PDB (Berman et al., 2000) detêm uma enorme riqueza de informações, armazenadas em uma estrutura computacional de banco de dados. Estas informações, são resultantes de milhares de horas de trabalho, de pesquisadores espalhados pelo mundo (Berman et al., 2007) e desta base de dados, é possível extrair informações que podem contribuir, de forma efetiva, impactando diretamente na vida das pessoas, a medida que pode, por exemplo, ajudar a entender o mecanismo de várias doenças e ainda na descoberta de fármacos.

Com relação aos trabalhos e pesquisa em Bioinformática, espera-se que as técnicas e os algoritmos desenvolvidos, sejam capazes de lidar com esta base de forma automatizada, o que pode não ser uma tarefa simples. Na seção seguinte, mostraremos pequenos problemas nos dados, que podem representar importantes desafios, na construção de métodos automatizados para análise desses dados.

2.6 Inconsistências e omissões

A etapa de preparação dos dados, é parte importante e necessária a qualquer trabalho científico em Bioinformática, visto que a adoção de uma base de dados inconsistente, pode impedir a realização do trabalho, ou mesmo direcioná-lo à conclusões erradas ou enviesadas, comprometendo toda a pesquisa. Nissink et al. (2002) descreve a necessidade de aplicação de critérios para identificar arquivos estruturais pouco confiáveis e que possam comprometer uma análise precisa. Segundo o autor, há dois tipos principais de erros encontrados nos arquivos do PDB: os fatuais e os estruturais (Nissink et al., 2002). Os fatuais são erros que estão ligados a omissões e resultam em estruturas incompletas, por exemplo, a falta de um átomo, um aminoácido ou mesmo uma cadeia inteira, são erros relacionados à imprecisão estrutural. Os erros relacionados à localização ou a existência dos átomos podem estar diretamente ligados à resolução da estrutura. Os erros estruturais são erros que colocam em dúvida a metodologia utilizada para obter a estrutura, uma vez que produzem arquivos estruturais errados.

Szabadka e Grolmusz (2007) sugere que a ausência de cadeias inteiras, ausência de átomos e ausência de resíduos, são as principais inconsistências encontradas nos arquivos do PDB. Apesar desta tarefa ser árdua, o autor sugere que é preciso tratar inconsistências como estas, antes do uso da base de dados em qualquer tipo de análise. Feldman et al. (2006), menciona a dificuldade em utilizar algoritmos para o estudo que realizou, estando a estrutura sem átomos de hidrogênio e contendo átomos com múltiplas ocupâncias, nas estruturas obtidas por Cristalografia por difração de raios X.

Desde de que o PDB foi criado, iniciativas para detecção e correção de inconsistências e imprecisões antes do processo de submissão para elevar a qualidade dos dados veem sendo criadas (Tagari et al., 2006; Boutselakis et al., 2003; Collaborative Computational Project, 1994). Estes processos tem surtido efeito e a melhoria das atuais estruturas são consideráveis. Entretanto, algumas omissões não dependem apenas da qualidade do pesquisador. O processo de cristalização e resolução da estrutura, influenciam diretamente na qualidade. Desta forma, as estruturas depositadas mais recentemente na base de dados tendem a ser de melhor qualidade.

Em seu estudo, Feldman et al. (2006) sugere que a falta de átomos de hidrogênio, dados que podem gerar imprecisão de uma coordenada atômica, podem acarretar atraso e imprecisão na análise de sítios ativos que reconhecem pequenas moléculas. Baseado neste estudo, construiu-se uma base de dados similar ao PDB denominada SMID, se

diferenciando pela produção de arquivos estruturais utilizando alta resolução e priorizando as estruturas com a menor quantidade de átomos ausentes.

2.7 Trabalhos relacionados

Há basicamente dois grupos de trabalhos que podem ser considerados relacionados ao nosso: o primeiro deles é composto por ferramentas, locais ou web, para busca e/ou normalização de arquivos PDB e o outro é composto por bibliotecas de códigos em diversas linguagens de programação cujo intuito é o mesmo, de obter e fazer diversos processamentos nos dados das estruturas obtidas.

Dentre as ferramentas disponíveis, o repositório proposto por Berman et al. (2000) denominado RCSB - *Research Collaboratory for Structural Bioinformatics*[3] oferece pesquisas ao repositório mundial que se destacam pela quantidade de opções proporcionadas com a integração de bases externas como SCOP (Conte et al., 2000), *Gene Ontology* (Ashburner et al., 2000), Pfam (Finn et al., 2014), e outros. Utilizando deste tipo de abordagem é possível mesclar critérios de pesquisas e selecionar um conjunto de dados abrangente e direcionado utilizando o operador lógico "AND". Aliado a estas pesquisas, estão a facilidades em baixar os arquivos do repositório mundial aliado a opções de visualizações instantâneas.

A biblioteca Open Babel (OLBoyle et al., 2011) trata se de uma ferramenta multiplataforma desenvolvida usando a linguagem de programação C++ e buscou atender uma demanda existente para interconectar os dados presentes entre os vários formatos de arquivos biológicos usados na modelagem molecular, química computacional e áreas afins. Esta biblioteca, juntamente com uma interface gráfica, está disponível para instalação e pode ser adquirido no site do projeto[4].

O PISCES (Wang e Dunbrack, 2005) oferece consultas à sua base de dados online para produção de listas de arquivos estruturais de acordo com consultas realizadas pelo usuário, que podem ser usadas para fazer o *download* destes arquivos nos diversos repositórios disponibilizados. A importância desta aplicação está justamente na possibilidade de usar critérios para diminuir a incidência de arquivos com muitas omissões na base de dados.

O software ProSA-web (Wiederstein e Sippl, 2007), disponível online, está focado em reconhecer erros em estruturas biomoleculares, experimentais ou teóricas, atuando nos processos de validação estrutural favorecendo o processo de análise do usuário. Este software, baseado no conhecimento existente na base de dados mundial, através de comparações estatísticas, utilizando Carbonos α (Cα), para possibilitar avaliar estruturas de baixa resolução.

A ferramenta proposta por Dolinsky et al. (2007), denominada PDB2PQR e também disponível através de um servidor web que atua, principalmente na validação de interações

[3] http://deposit.pdb.org/validate/
[4] http://openbabel.org/

eletrostáticas, oferece funções de adição de átomos faltantes na estrutura da biomolécula com o objetivo de preparar estruturas tridimensionais para utilização em ferramentas computacionais como softwares de simulações biomoleculares, estimando-se estados de protonação biomolecular, de maneira consistente com ligação de hidrogênio favorável, atribuindo carga e raio usando parâmetros de uma variedade de campos de força gerando uma 'PQR' (um formato *PDB-like*). Neste novo formato, compatível com várias aplicações computacionais, as colunas de ocupação e do fator de temperatura, substituídos com carga 'Q' e raio 'R', respectivamente.

Paralelamente ao desenvolvimento dessas ferramentas, vários colaboradores tem se empenhado em evoluir as bibliotecas desenvolvidas em diversas linguagens de programação, para que possam lidar com especificidade de dados biológicos, implementando classes e métodos para facilitar o tratamento desses dados, como Python (Cock et al., 2009), Java (Holland et al., 2008), R (Grant et al., 2006) dentre outras (Gajda, 2013). Este tipo de abordagem é bastante flexível e poderosa, mas atrelado ao alto custo em dominar lógica de programação e a linguagem que se deseja utilizar. Entretanto, alguns pesquisadores tendem a usar tecnologias que demandem menos tempo de aprendizado, para lidar com a própria tecnologia, buscando minimizar a chamada curva de aprendizado, no intuito de se dedicar exclusivamente ao desenvolvimento de seu próprio trabalho e análises.

Capítulo 3

Metodologia

Este trabalho propõe uma ferramenta para aquisição, manipulação, filtragem e conversão de arquivos PDB. A partir desta seção, toda vez que o termo *arquivo PDB* for mencionado, o que se pretende é referenciar qualquer arquivo, de biomolécula, que mantenha compatibilidade com a versão 3.3[1]. Caso seja de interesse verificar a compatibilidade de algum arquivo, com as especificações corretas, o site RCSB - *Research Collaboratory for Structural Bioinformatics*[2] (Berman et al., 2000) fornece uma boa ferramenta para validação desta estruturas. O Software produzido neste trabalho foi denominado de PDBest® - PDB Enhance Structure Toolkit, e as cores de sua logomarca remetem às cores da bandeira nacional do Brasil.

Definimos brevemente o que entendemos por cada um dos termos que por sua vez definem as funcionalidades da ferramenta desenvolvida:

- Por **aquisição**, entende-se o processo de captura dos arquivos estruturais. Estes arquivos podem ser advindos de uma base de dados local (um diretório no sistema de arquivos do computador do usuário) ou podem ser obtidos através da base de dados do PDB através do repositório RCSB[3] (Berman et al., 2000).

- Por **manipulação**, entende-se o processo de organização dos arquivos em diretórios específicos, definidos pelo usuário através do PDBest.

- Por **edição**, entende-se o processo de alteração/inserção das informações dos arquivos de entrada e que, são mudanças que, necessariamente, afetam a estrutura da molécula.

- Por **filtragem**, entende-se o processo de retirar da estrutura, informações não relevantes, para um dado contexto de pesquisa, desde de que não afete a estrutura ou fragmento original.

[1] http://www.wwpdb.org/documentation/file-format
[2] http://deposit.pdb.org/validate/
[3] http://www.rcsb.org

- Por **conversão**, entende-se o processo de mudança na estrutura do arquivo, para outros formatos, que não o PDB.

3.1 Desenvolvimento

3.1.1 Linguagem utilizada

Criadas para intermediar a especificação de tarefas entre as pessoas e computador, as linguagens de programação estão presentes na computação em grande variedade. Cada uma delas com um conjunto de instruções específico que informará ao computador o que, e quando, executar uma determinada tarefa.

Muito embora a variedade de linguagens seja grande, algumas delas tem grande notoriedade em Bioinformática, não só por oferecerem bibliotecas específicas para tratar dados biológicos, como BioJava (Holland et al., 2008) e Biopython (Cock et al., 2009), mas também por se adaptarem melhor a este contexto, oferecendo facilidades em lidar com estruturas de dados do cotidiano dos bioinformatas.

Muitas vezes, buscando agilidade de desenvolvimento, o desempenho é deixado de lado e atividades simples, como o processamento de arquivos texto, podem demorar mais que o esperado. Para manter um tempo de processamento aceitável ao desempenhar as atividades propostas neste trabalho, o PDBest (Gonçalves et al., 2015) precisou tratar os dados com o máximo de eficiência possível. Esta também foi a preocupação de Fourment e Gillings (2008) ao desenvolver um estudo que comparou 6 das linguagens, que ele considerou mais relevantes em Bioinformática, oferendo um quadro comparativo que contribui para subsidiar a escolha de uma determinada linguagem de programação. Ele comparou C, C++, C Sharp, Java, Perl e Python. A Figura 3.1 mostra uma comparação que leva em conta a velocidade de execução de um conhecido algoritmo de alinhamento de sequências, o BLAST. Ainda que o PDBest não execute tarefas de alinhamento de sequências, nota-se que há uma grande diferença entre a velocidade de processamento destas linguagens, ficando com C e C++ os melhores resultados, o que este trabalho de não mostrou é a curva de aprendizado destas linguagens que, na maioria das vezes, é inversamente proporcional ao desempenho. Ainda que o PDSBest não execute alinhamentos, as comparações de Fourment e Gillings (2008), apontam resultados mais eficientes para as linguagens C, C++, C# e JAVA quanto ao tempo de processamento e utilização de memória, dando ainda mais segurança na escolha da linguagem C++ para implementação do PDBest.

Durante a implementação de um software gráfico, é preciso lidar com interações do usuário com a interface, através de eventos de mouse, teclados e também com outros atributos relacionados a interface gráfica. Como esse processo é complexo, já existem *frameworks* que encapsulam essas tarefas, provendo facilidades na implementação de interfaces gráfica para os principais sistemas operacionais. Como trabalhamos com C++, decidimos usar o Qt.

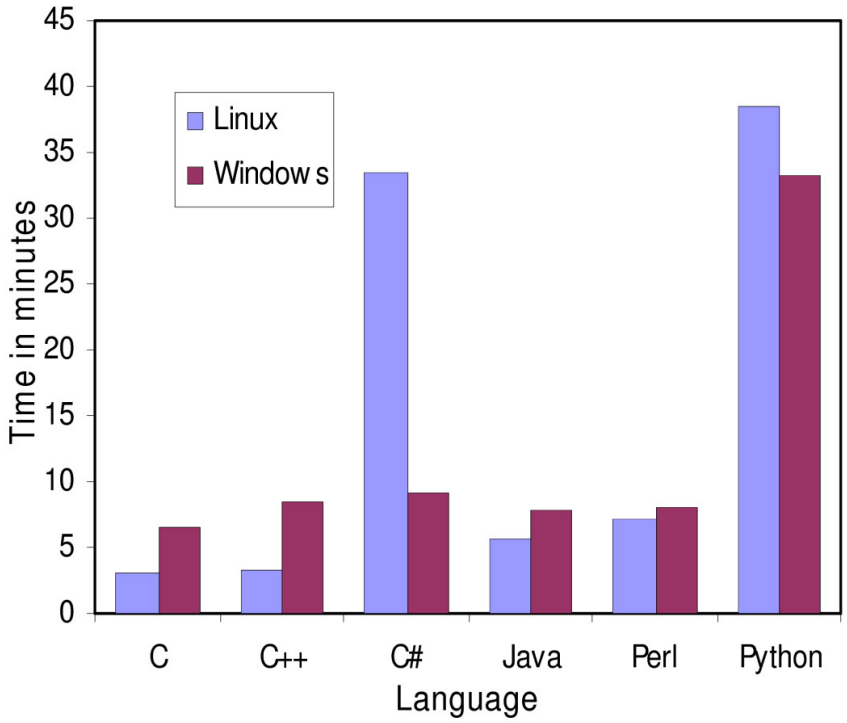


Figura 3.1: Comparação da velocidade na execução do BLAST em diversas linguagens de programação

Fonte: Fourment e Gillings (2008)

Criado por Haarvard Nord e Eirik Chambe-Eng, cofundador da empresa Trolltech e disponível publicamente a partir de maio de 1995, Qt já em seu início foi concebido para ser multiplataforma. Cerca de um ano após o seu lançamento a agência espacial europeia se tornou cliente do Qt e seis meses depois a versão 1.0 foi lançada. Nos anos seguintes, projetos como K Desktop Environment - KDE[4] Summerfield (2010), apareceram e versões para Windows, MacOS X, Unix e Linux. Em meados de 2005, uma das versões mais utilizadas foi lançada, a versão 4.0, contando com 500 classes e mais de 9000 funções já implementadas.

Adquirida pela Nokia, chamando-se Qt Software, foi disponibilizado sob as licenças GPL[5], LGPL[6] e comercial (paga). Atualmente a Nókia abandonou o desenvolvimento do Qt e a empresa Qt Software foi adquirida pela Digia, uma colaboradora do projeto Qt, que está um sua verão 5.4, oferecendo suporte para implementar aplicações, Windows, Windows RT, Mac OS X, iOS, Linux e Android (Summerfield, 2010).

[4] Comunidade de software livre para produção de aplicativos multiplataforma.
[5] *General Public License* (Licença Pública Geral)
[6] *Lesser General Public License* (Licença Pública Geral Menor)

3.1.2 Sistemas operacionais escolhidos

Um sistema de computação é constituído basicamente por hardware e software. Por hardware podemos entender os circuitos eletrônicos (processador, memória, portas de entrada/saída, etc.) e periféricos (teclados, mouses, discos rígidos, unidades de, CD ou DVD, dispositivos USB, etc.). Como Software, genericamente podemos entender como programas que se comunicam com o hardware.

Os softwares podem, grosseiramente serem divididos em dois tipos: programas de sistema, que gerenciam as operações de um computador em si e programas aplicativos, que realizam a real atividade desejada pelo usuário. Dentre os programas de sistema está o mais básico, e talvez o mais complexo de todos, o Sistema Operacional. O Sistema Operacional tem a tarefa de controlar os recursos disponíveis e prover aos programas aplicativos, os recursos necessários para o seu funcionamento (ex: Linux, Windows, MAC OS X, Android, Unix) (Tanenbaum e Woodhull, 2008).

É possível encontrar nos computadores atuais duas arquiteturas: de 32 ou 64 bits. Essas arquiteturas estão relacionadas a quantidade de bits[7] que se pode utilizar em uma única operação. Resumidamente falar desta arquitetura equivale dizer que processadores de 32 bits têm a capacidade de processar "palavras"(sequência de bits) de até 32 bits, enquanto os de 64 bits podem trabalhar com "palavras"de até 64 bits, ou seja, o dobro de informações. Assim, a capacidade de um hardware do gênero poder trabalhar com uma quantidade maior de bits, entretanto, isso não influenciará diretamente em sua velocidade de operação, mas sim, em um melhor desempenho geral do conjunto.

Muito embora o nome Linux seja referenciado de forma única, por se tratar de software livre, várias distribuições se propagam. Uma distribuição Linux contém um núcleo Linux aliado a pacotes próprios, fazendo com que cada distribuição seja independente, mesmo que haja interseção em alguns pontos. Isso faz com que, ao desenvolver um software para Linux, informações como versão do sistema e distribuição seja levada em consideração. O que não acontece com os sistemas proprietários, como o caso dos sistemas fornecidos pela Apple e Microsoft, respectivamente OS X e Windows.

Inicialmente foram escolhidos três Sistemas Operacionais para serem suportados pelo PDBest: OS X (Apple), Windows (Microsoft) e Linux (Open Source). Para definir corretamente as versões adotadas neste projeto, foram utilizadas versões estáveis no início da implementação, a saber Windows 7, OS X 10.9 e no caso do Linux, foram escolhidas duas distribuições, o Debian e o Ubuntu em suas versões 64 e 32 bits conforme Tabela 3.1. Especialmente para o Ubuntu, que oferece duas versões oficiais a cada ano, a escolha que pareceu mais sensata foi escolher as distribuições LTS [8] que oferece 3 anos de suporte e é disponibilizada a cada dois anos, oferecendo portanto 13 versões dos sistemas operacionais mais utilizados.

[7] Menor unidade de informação que pode ser armazenada ou transmitida.

[8] *Long Term Support*

Tabela 3.1: Sistemas operacionais suportados

Sistemas Suportados			
Windows	Windows 7		
	Windows XP		
Mac OS X	MAC OS X v10.9 "Mavericks"		
	MAC OS X v10.10 "Yosemite"		
Linux	Ubuntu LTS	10.04	32 bits
		12.04	32 e 64 bits
		14.04	32 e 64 bits
	Debian	Wheezy	32 e 64 bits
		Jessie	32 e 64 bits

3.1.3 Implantação

Resumidamente, Implantação de Software é uma das fases de seu ciclo de vida e corresponde à transição do software para produção. Segundo Rezende (2005), o processo de implantação pode ser interpretado como um processo comum a todo software, o qual tem que ser customizado de acordo com as especificidades e características de cada software. Neste contexto o software desenvolvido precisa ser codificado, compilado e instalado.

- **Codificação:** Trata-se do processo de escrita de código e foi o que tomou a maior parcela de tempo neste trabalho. Juntamente com o *framework*, é disponibilizada um Ambiente Integrado de Desenvolvimento - IDE[9] que foi usado para editar os códigos fontes.

- **Compilação:** Trata-se basicamente de traduzir a linguagem utilizada, no caso C++, para linguagem de máquina. Como optamos por uma abordagem multiplataforma, cada máquina tem sua linguagem e é preciso realizar uma compilação para cada sistema suportado. Para isso necessitamos de uma instância de cada sistema operacional com instalação de todo o ambiente de desenvolvimento e suas bibliotecas. Esse é um dos principais motivos que nos levaram a definir um escopo prévio e um número limitado de distribuições.

- **Instalação:** O processo de instalação consiste em colocar o software em funcionamento e em condições de funcionamento, incluindo bibliotecas e arquivos necessários.

3.2 Funcionalidades

Esta sessão, descreve as funcionalidades implementadas no PDBest, agrupando-as de acordo com o que há em comum entre elas, a saber as já mencionadas funcionalidades estão agrupadas em Aquisição, Filtragem, Edição, Análise e Reprodutibilidade.

[9] *Integrated Development Environment*

3.2.1 Aquisição

Por aquisição entende-se a indicação dos arquivos que serão processados, o qual podem ser originados de bases de dados locais, quando o usuário mantém sua base de dados proprietária, ou retirados da base de dados PDB internacional, através do repositório RCSB[10].

A Figura 3.2 traz a tela inicial do PDBest. Ela exemplifica a busca por estruturas com resoluções entre 0 e 2 Ångstrons e que sejam classificadas pela base de dados *Structural Classification of Proteins - SCOP* (Conte et al., 2000), como "Globin-like (core: 6 helices; folded leaf, partly opened)"e que tenham similaridade de sequência superior a 30%.

Na lateral esquerda da mesma figura é possível identificar a principal divisão do fluxo de trabalho do PDBest, que são a inserção de dados, as configurações dos filtros e o processamento das estruturas.

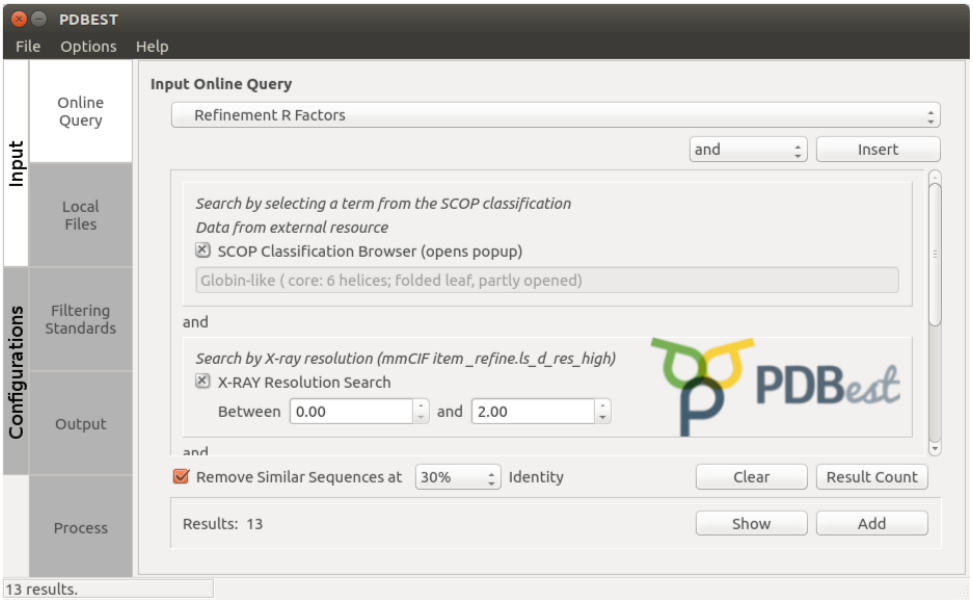


Figura 3.2: Tela inicial

3.2.1.1 Aquisição Online

Caracterizada pela aquisição/*download* dos dados através do repositório RCSB (Berman et al., 2000)através de um *web-service* disponibilizado pelo próprio site, é a forma mais indicada para o carregamento de uma estrutura tridimensional no PDBest. Assim, garantindo a conformidade com o formato correto do arquivo. Além de ser possível usar as regras de seleção já estabelecidas pelo RCSB, para filtrar as estruturas retornadas, assim como o exemplo mostrado na Figura 3.2.

A Figura 3.3(a) mostra, no topo, uma caixa de seleção que, ao ser acionada, como na Figura 3.3(b), fornece opções para o refinamento na busca de aquivos, semelhantes às

[10]http://www.rcsb.org

apresentadas na Figura 3.2, estas opções referem-se às mesmas opções apresentadas pelo site RCSB[11] para que o usuário se sinta familiarizado com uma interface já bastante utilizada. Uma vez estabelecidos os critérios de busca na base de dados, é possível identificar quantas estruturas serão retornadas, e então reavaliar os parâmetros utilizados.

O exemplo mostrado na Figura 3.2, ilustra bem a utilização do operador lógico "AND"entre as consultas, entretanto também é possível usar o operador lógico "OR"para combinar as consultas.

Dado duas consultas à base de dados do PDB, uma consulta A e uma outra consulta B, distintas entre elas e unidas pelo operador lógico "AND", terá como resultante somente os arquivos presentes em ambas as consultas. Para o caso destas mesmas consultas estarem unidas pelo operador lógico "OR", o conjunto de arquivos resultantes será a soma dos elementos retornados entre a consulta A e a consulta B. A principal diferença prática entre elas é a oportunidade de criar, no caso do "AND", consultas restritas e no caso do "OR", consultas abrangentes, e ainda sim selecionando apenas um conjunto de interesse na base de dados.

3.2.1.2 Aquisição local

A possibilidade de carregar arquivos locais é importante, a medida que cada usuário possa trabalhar com arquivos já alterados, por algum método com características próprias, ou estruturas ainda não submetidas, ou mesmo as modeladas por homologia.

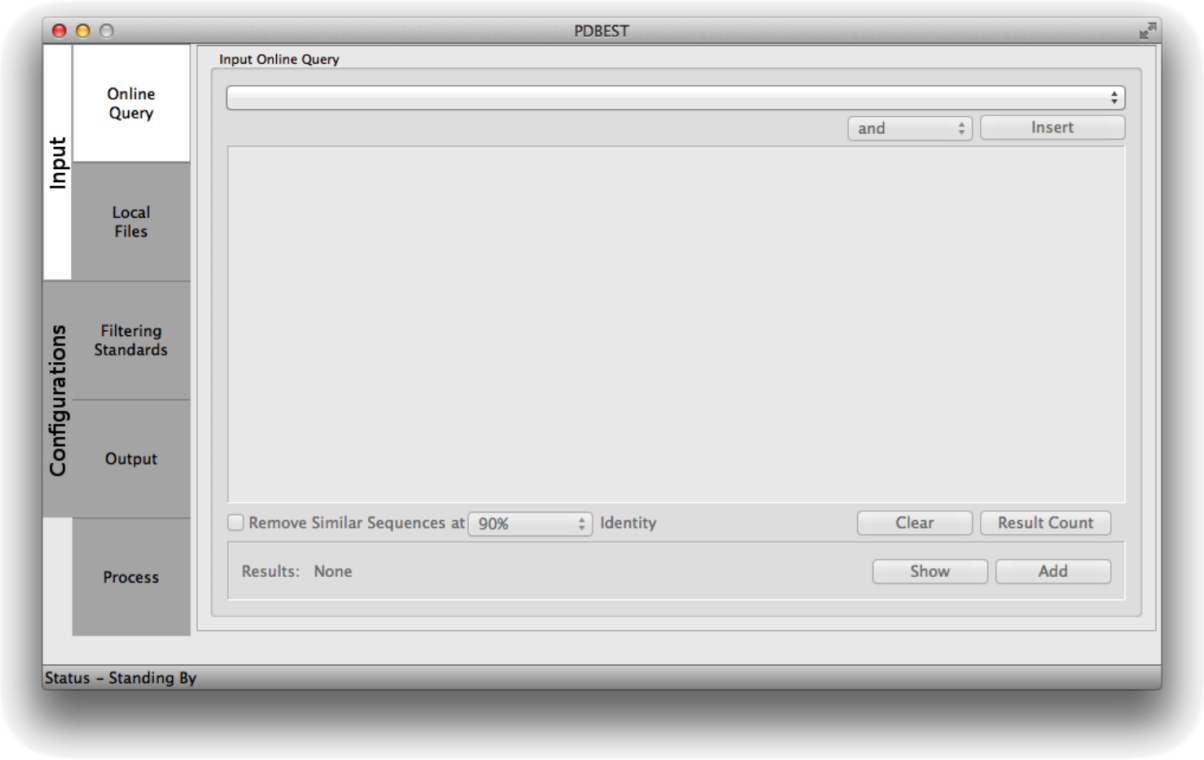
Para usar arquivos locais, é necessário apenas indicar o local onde o arquivo se encontra, ou a pasta onde se quer buscar os arquivos. Isso pode ser feito através de indicação explícita, ao selecionar a pasta exata onde o arquivo se encontra ou implícita ao carregar um arquivo no formato .txt com a localização de cada arquivo. A Figura 3.4 mostra como pode ocorrer a indicação dos arquivos localmente para processamento, e a Tabela 3.2, gerada através ao executar o comando [ls -d $PWD/*] em um sistema operacional Linux, mostra como pode ser a estrutura de um arquivo usado para indicar diferentes locais de armazenamento.

Recentemente, o consórcio internacional padronizou a forma de submissão das estruturas para o formato mmCif, já que o formato PDB não comporta estruturas maiores que 62 cadeias e/ou 99.999 átomos.

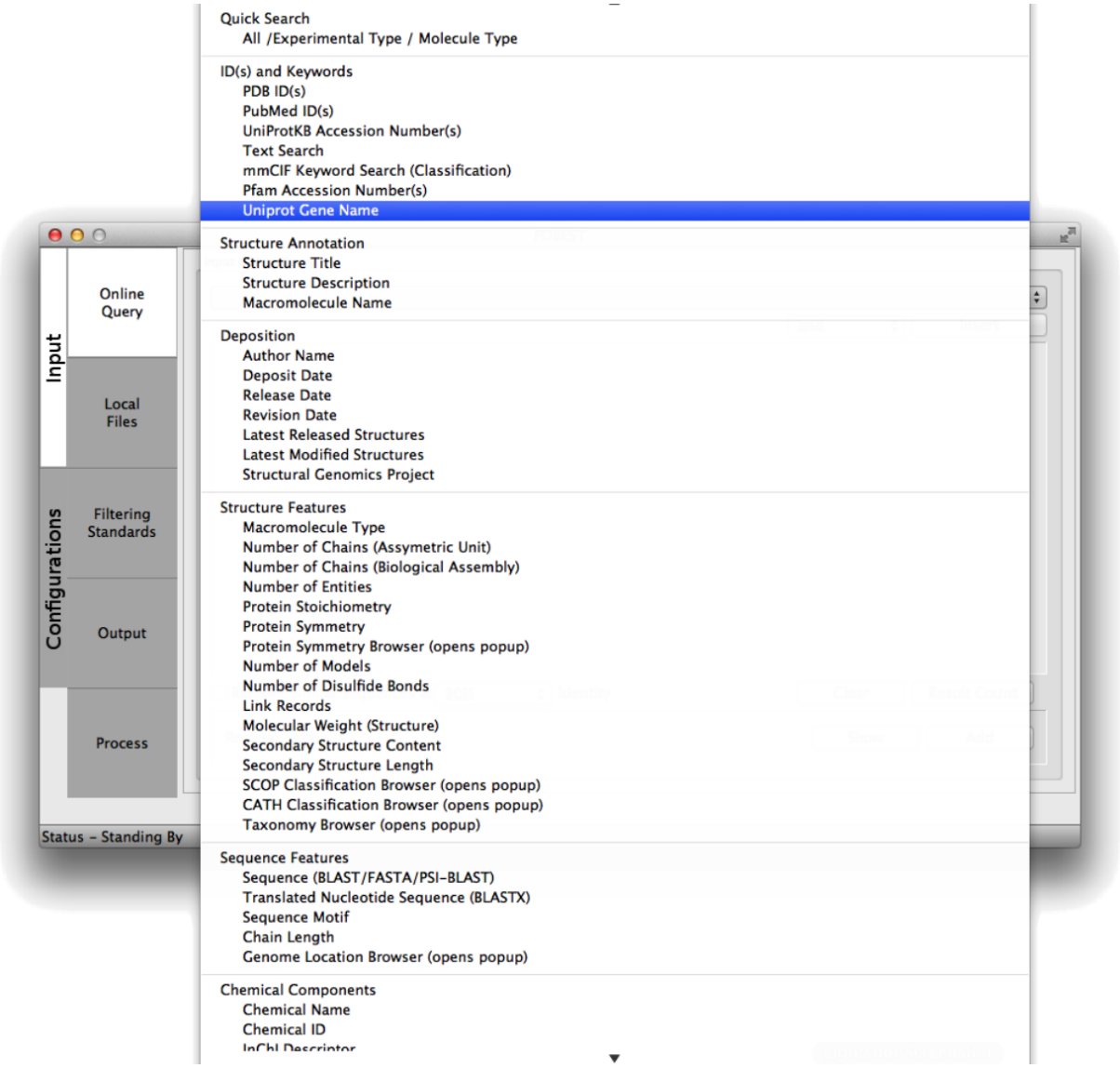
Desta forma são dois os formatos aceitos pelo PDBest:

- **PDB:** Trata-se do arquivo proposto por Bernstein et al. (1977).
- **mmCif:** Para manter o sistema de submissão de arquivos atualizados. Ao submeter um arquivo no formato mmcif, automaticamente o PDBest converte este arquivo usando funcionalidade do Open Babel (OLBoyle et al., 2011). Sendo assim também recomendamos o uso do formato proposto por Bernstein et al. (1977) (PDB).

[11] www.rcsb.org



(a) Forma contraída



(b) Forma expandida

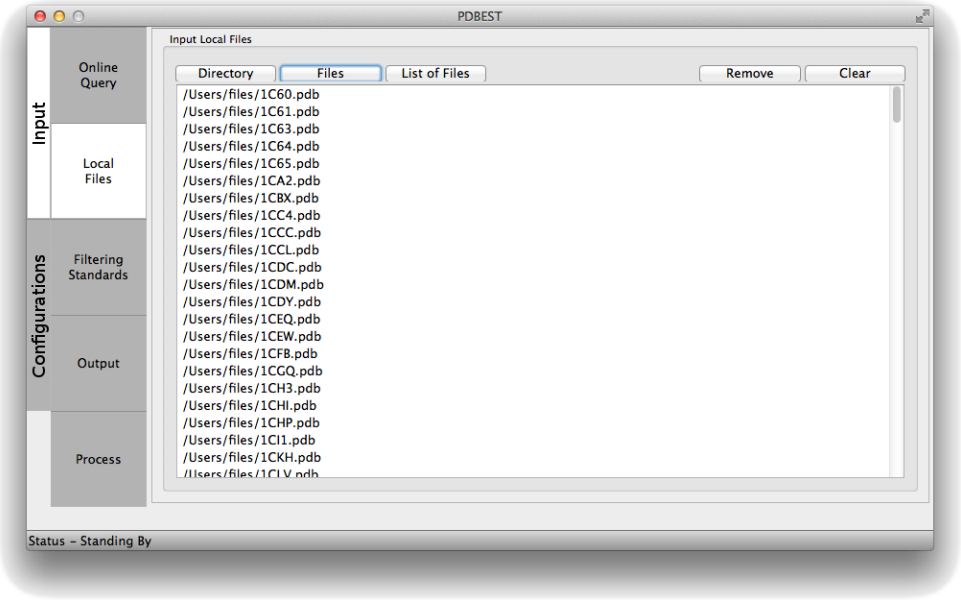


Figura 3.4: Tela de aquisição da base de dados através de inserções locais

Tabela 3.2: Exemplo de arquivo com indicação de locais diversos.

/user/downloads/PDBest/Downloads/files/pasta1/1E5Z.pdb
/user/downloads/PDBest/Downloads/files/pasta2/1E97.pdb
/user/downloads/PDBest/Downloads/files/pasta2/1EAM.pdb
/user/downloads/PDBest/Downloads/files/pasta3/1EB2.pdb
/user/downloads/PDBest/Downloads/files/pasta4/1EB4.pdb
/user/downloads/PDBest/Downloads/files/pasta5/1EDH.pdb

3.2.2 Filtragem

O termo filtragem está relacionado com as operações realizadas nos arquivos submetidos, as quais não gerem modificações das estruturas tridimensionais subsequêntes, mesmo que após um ciclo de processamento a estrutura original seja fracionada ou particionada.

Um ciclo de processamento compreende o processo de escolha das estruturas, seleção das operações a serem realizadas e a aplicação destas operações nos arquivos selecionados. A Figura 3.5 mostra como é um ciclo de processamento do software. Este processo pode ser refinado tantas vezes quanto o usuário ache necessário modificar os parâmetros.

A Figura 3.6 apresenta um conjunto de filtros implementados no PDBest, entretanto, as funcionalidade de adição e remoção de hidrogênio e conversão de formato, não coincidem com os conceitos apresentados neste item, os quais serão abordados mais adiante. As demais opções de filtragem presentes no PDBest estão diretamente relacionadas com as seções apresentadas na Tabela 2.4 onde cada utilizador pode conservar ou remover blocos de informações desnecessários a um determinado contexto.

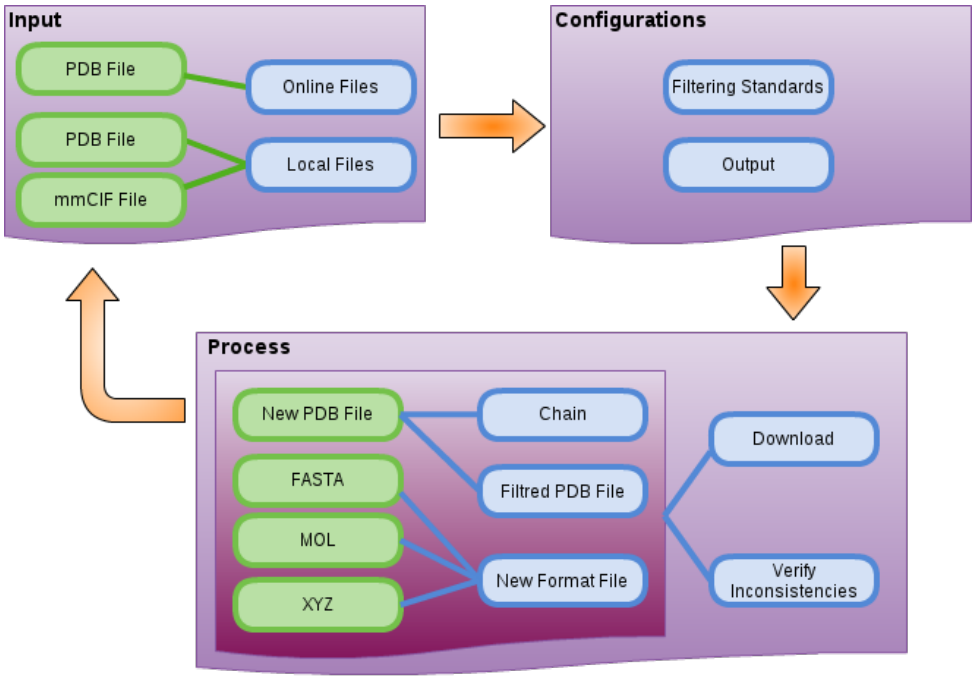


Figura 3.5: Ciclo de processamento do PDBest

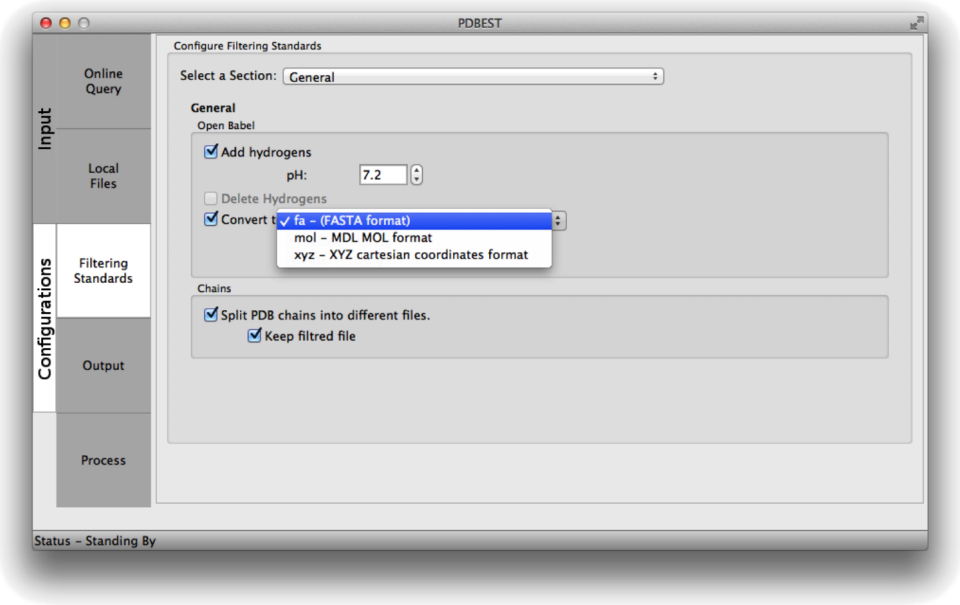


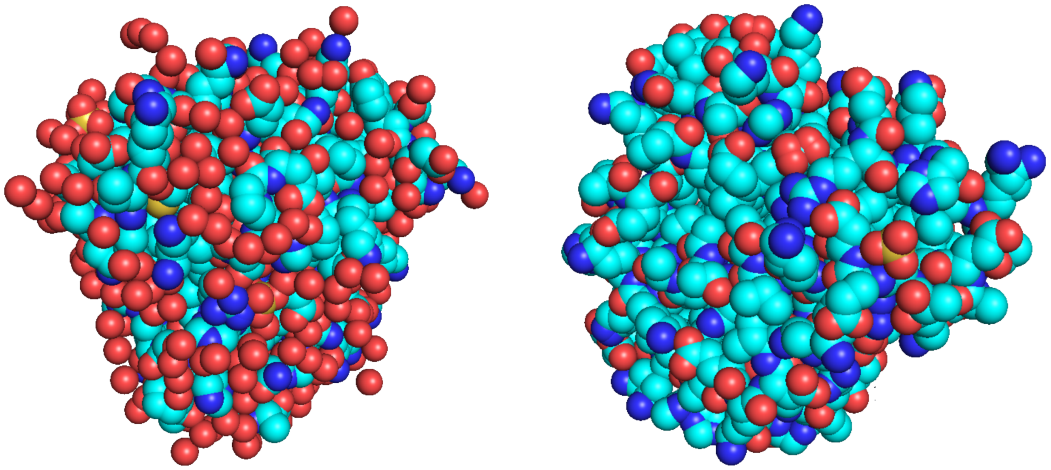
Figura 3.6: Filtragem

3.2.3 Edição

O termo edição descreve operações realizadas pelo PDBest que, necessariamente, resultem em mudanças na estrutura tridimensional, ou que corrijam alguma inconsistência encontrada no arquivo. As principais edições que o PDBest traz são a remoção de átomos de hidrogênio, a renumeração de átomos e resíduos, a adição de átomos de hidrogênio realizada pelo software Open Babel (OLBoyle et al., 2011), remoção de moléculas de água e a remoção de múltiplas ocupâncias.

A Figura 3.7 mostra o resultado de uma edição realizada retirando as moléculas de

água da estrutura.



(a) original (b) Sem moléculas de água

Figura 3.7: Remoção de moléculas de água na estrutura 1BZR - Mioglobina

3.2.4 Análise

Após carregar os arquivos a serem processados, mesmo antes de selecionar as filtragens e as edições a serem realizadas, é possível realizar uma análise global de todo o conjunto de dados realizando um pré-processamento para identificar 4 das principais inconsistências que podem ser encontradas nos arquivos pdb. São elas:

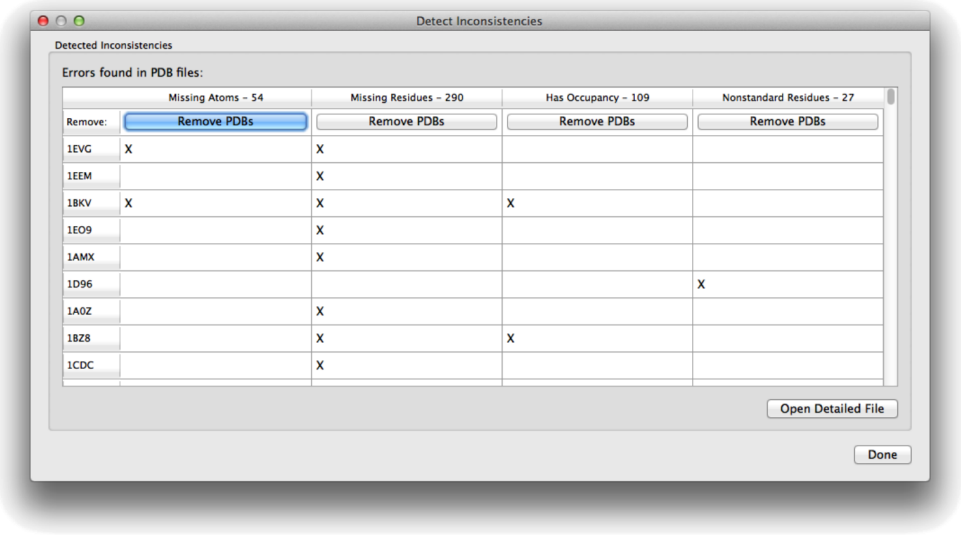
- **Átomos Ausentes:** A numeração dos átomos de um arquivo estrutural é global e é atribuído a cada átomo um número sequencial que o irá identificar na estrutura. Uma falha acontece quando esta sequência é interrompida e uma lacuna entre esses números é encontrada. Isso é diferente com estruturas obtidas por Ressonância Nuclear Magnética, nas quais essa numeração é reiniciada para cada modelo apresentado.

- **Resíduos Ausentes:** Assim como os átomos, cada resíduo recebe uma numeração e a falha acontece quando ocorre um intervalo na sequência destas numerações.

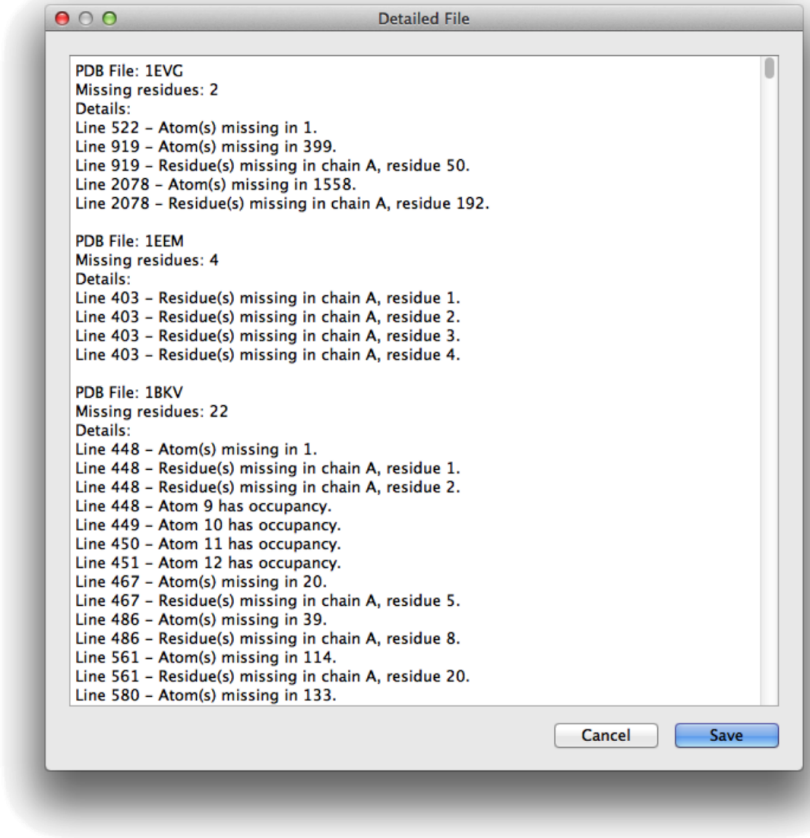
- **Múltiplas Ocupâncias:** Durante a realização do método experimental que dará origem a estrutura, pode não ser possível definir com precisão onde uma átomo se encontra, pois ela pode ser encontrada em mais de uma conformação diferente. Quando isso ocorre, duas ou mais linhas no arquivo estrutural, irão descrever o mesmo átomo, diferenciando-se pelo campo reservado a ocupância e às coordenadas atômicas, que conterão o valor correspondente à porcentagem do átomo que está naquelas coordenadas (x,y,z) e as tais coordenadas.

- **Resíduos Não-Padrão:** Ocorre quando a estrutura contém um resíduo que não está presente dentre os resíduos considerados padrão, que são os 20 aminoácidos mais comumente encontrados nos seres vivos e listados previamente na Figura 2.3.

A Figura 3.8(b) mostra um relatório detalhado gerado após o pré-processamento de um conjunto de dados selecionado aleatoriamente. Nesta fase é possível salvar o resultado do processamento e visualizar as inconsistências presentes em cada arquivo do conjunto de dados, além de avaliar sua eventual exclusão ou direcionar o conjunto de filtros ou edições a serem realizadas. Entretanto, durante esta fase de pré-processamento, também é possível ser bastante conservador removendo facilmente todos os arquivos que apresentem uma determinada inconsistência, como mostrado na Figura 3.8(a)



(a) Análise global



(b) Análise detalhada

Figura 3.8: Detalhamento da análise de conjunto de dados

3.2.5 Conversão

Através do software Open Babel (OLBoyle et al., 2011), o PDBest pode converter as estruturas antes ou depois da aplicação dos filtros e edições, para três formatos distintos como mostra a Figura 3.6:

- **FASTA:** Descreve a sequência de aminoácidos que representam uma estrutura proteica. Essencialmente cada aminoácido é representado por uma letra:
- **MOL:** Usado para codificar estruturas, subestruturas e conformações químicas como tabelas de conexões baseadas em texto.
- **XYZ:** Descreve apenas as coordenadas atômicas para cada átomo da estrutura.

A Tabela 3.3 traz um pequeno fragmento de cada arquivo, na mesma ordem em que são apresentados no texto.

Tabela 3.3: Fragmento dos arquivos convertidos 1BZR - Mioglobina

VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKS	N	-14.67100	-3.87900	-0.18600	M	V30	1	N	-14.671	-3.879	-0.186	0
HPETLEKFDRFKHLKTEAEMKASEDLKKHGVTVLT	C	-15.67000	-3.40100	0.74800	M	V30	2	C	-15.67	-3.401	0.748	0
ALGAILKKKGHHEAELKPLAQSHATKHKIPIKYLE	C	-15.90200	-4.46500	1.82700	M	V30	3	C	-15.902	-4.465	1.827	0
FISEAIIHVLHSRHPGDFGADAQGAMNKALELFRK	O	-15.81700	-5.67000	1.62800	M	V30	4	O	-15.817	-5.67	1.628	0
DIAAKYKELGYQGXXX	C	-16.99400	-3.00500	0.05900	M	V30	5	C	-16.994	-3.005	0.059	0

Fonte: http://rcsb.org

3.2.6 Reprodutibilidade

Permite o compartilhamento das funcionalidades utilizadas no processamento dos dados, através de dois tipos de arquivos. Um, denominado de arquivo de protocolo (.prt), contendo as consultas online e todos os filtros e edições aplicados em um ciclo de processamento e outro, denominado de arquivo de filtro (.fil), contendo apenas os filtros e as edições utilizadas. Em nenhum dos casos arquivos locais são considerados por suas características peculiares e pela impossibilidade de reprodução dos experimentos com arquivos de outros usuários.

Esta funcionalidade é particularmente especial, pois permite a interação entre grupos de pesquisa ou mesmo pesquisadores mais experientes, compartilhem os seus dados para que sejam reprodutíveis. Um outra vantagem desta técnica é a facilidade do pesquisador em armazenar seus próprios resultados salvando o protocolo utilizado ou reutilizando-o em diversas bases de dados a serem estudadas.

Após o usuário selecionar quais arquivos serão trabalhados, quais os filtros serão adotados e quais edições serão realizadas, poderá acessar o *menú* principal e armazenar este ciclo de trabalho. Neste procedimento deverá indicar um local para armazenar o novo arquivo de protocolo/filtro que poderá ser gerado a cada ciclo de processamento3.5 . Do mesmo modo, também poderá acessar este mesmo *menú* para carregar um protocolo/filtro já salvo anteriormente.

Não estão incluídos nesta funcionalidade os arquivos locais processados, bem como os arquivos que foram retirados na fase de detecção de inconsistências.

3.2.7 Processamento de alto desempenho

Os arquivos manipulados pelo PDBest são essencialmente texto estruturado. As operações de leitura e escrita de arquivos locais estão entre os mais custosos computacionalmente e para cumprir como o requisito de manter o tempo de processamento o mais reduzido possível, foi necessário efetuar tarefas em paralelo e neste ponto o uso do *framework Qt* foi especialmente importante. O *framework Qt* possui bibliotecas que facilitam a implementação de tarefas em paralelo. Através de bibliotecas como *Qt Concurrent*, foi possível implementar multiprocessamento a nível de tarefas (*Threads*).

A biblioteca Qt Concurrent fornece uma interface de alto nível, para implementar softwares multitarefas (*multi-threaded*), usando programação de alto nível sem que o uso de estruturas de segmentações primitivas (de baixo nível), como por exemplo o uso de semáforos. Softwares escritos usando esta biblioteca ajustam automaticamente o número de tarefas de acordo com o número de núcleos de processamento disponíveis. Em outras palavras, significa que implementações atuais continuarão a escalar processos em paralelo, em sistemas *multi-core* no futuro.

Tarefas como downloads, a aplicação de filtros, puderam ser paralelizadas utilizando os benefícios da biblioteca mencionada.

Gerada através de um software para monitoramento de processos e recursos em um computador (HTOP[12]), a Figura 3.9 mostra a utilização dos recursos disponíveis em um computador com 4 núcleos e 6 GB de memória física. Simplesmente para gerar a imagem, um conjunto de dados contendo 2.000 arquivos foi submetido, e as tarefas de separação do arquivos em suas respectivas cadeias e a seleção, apenas de átomos da cadeia principal, portanto a remoção de tudo que não se enquadra neste critério.

Podemos ver que os processos desencadeados pelo PDBest estão consumindo praticamente todo o recurso de processamento existente na máquina. Pensando em hardware modestos e suas múltiplas funções, também foi implementado um módulo que permite desativar o recurso *Multi-thread*, deixando o processador livre por mais tempo, entretanto este processo eleva muito o tempo de processamento.

[12]HTOP

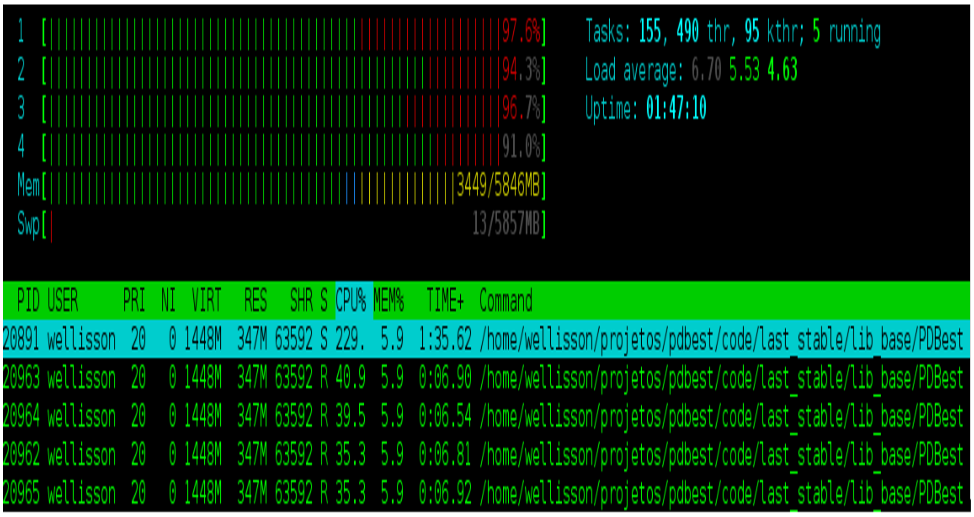


Figura 3.9: Processamento paralelo

Capítulo 4

Resultados e discussões

Neste capítulo descrevemos os principais resultados obtidos neste trabalho. São eles: o desenvolvimento do software PDBest; a construção de um site para disponibilização do mesmo; uma documentação do sistema sob forma de um manual e estudos de caso de uso da ferramenta, tudo isso publicado em um artigo científico em periódico. Por fim, mostramos uma breve discussão sobre resultados de comparações do nosso sistema com trabalhos relacionados apresentados anteriormente.

4.1 O Software

O software projetado neste trabalho foi registrado no Instituto Nacional de Patentes - INPI conforme Gonçalves et al. (2014) e a marca PDBest está em processo de homologação, pois é de praxe cerca de dois anos para se registrar uma marca. A marca contém o nome e marca, visual, desenvolvida para o PDBest conforme Figura 4.1

Figura 4.1: Marca PDBest

Fonte: Gonçalves et al. (2014)

4.2 O Site

O site[1], desenvolvido em PHP[2] e Bootstrap[3], elaborado para comunicação bidirecional com a comunidade através das informações apresentadas lá e por envio de mensagens ao site conforme a Figura 4.2 apresenta a tela de contato, que fornece exige duas informações obrigatórias: nome e e-mail.

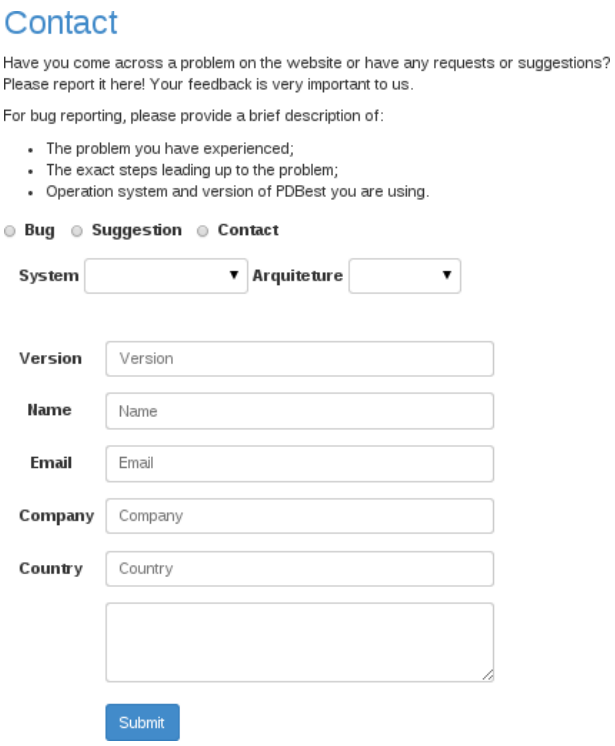


Figura 4.2: Página de contato

Para fazer o download do PDBest o usuário precisa fazer um cadastro prévio simples, informando obrigatoriamente nome e e-mail, os outros campos são opcionais. A Figura 4.3, mostra a simplicidade para fazer o download dos executáveis e suas bibliotecas. Os dados sobre o usuário são opcionais e, quando fornecidos, são armazenados em uma base de dados para posterior análise.

Os dados coletados na tela de Contato 4.2 e na tela de *Download* 4.3, são armazenados em uma única tabela de um banco de dados relacional. A Tabela 4.1 mostra como a base de dados foi estruturada.

O site também abriga quatro estudos de caso reais e o manual de utilização que será discutido posteriormente.

[1] www.pdbest.dcc.ufmg.br

[2] PHP - linguagem de programação, comumente usada para o desenvolvimento de aplicações web

[3] Bootstrap - *Framework* para geração de conteúdo adaptativo para WEB

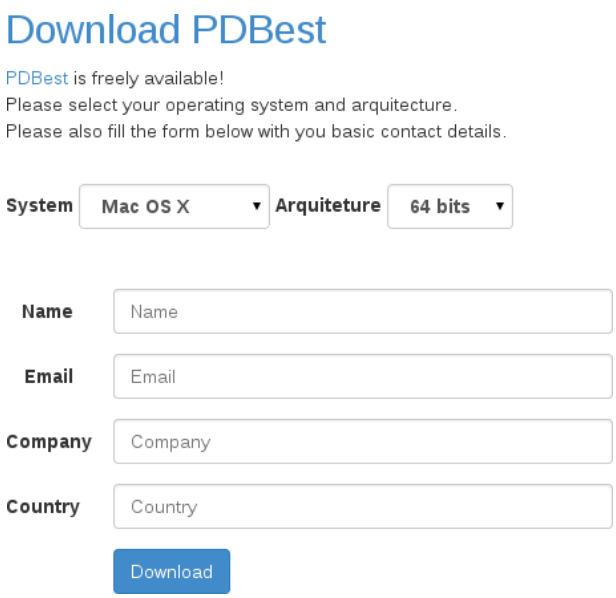


Figura 4.3: Página de download

Tabela 4.1: Banco de dados para armazenamento dos dados de usuário

USER	
ID	Identificador Sequencial da base de dados
NOME	Nome passado pelo usuário no momento do download
EMAIL	E-mail passado pelo usuário no momento do download
COMPANY	Atributo opcional informado pelo usuário
SISTEMA	Coletado após a escolha do sistema operacional
PLATAFORMA	Coletado após a escolha da plataforma, 32 ou 64 bits
VERSÃO	O sistema coleta a ultima versão estável do software
TIPO	Tipo de contato realizado pelo usuário
MENSAGEM	Mensagem deixada pelo usuário
IP	O sistema coleta o ip de origem da conexão
DATAHORA	Data e Hora que o usuário faz o download do software

4.3 O Manual

O manual, disponível no site na seção *Help Center*, está anexo a este trabalho no Apêndice 5.1.2. De forma simples, mas completa, cada tela extraída do software é explicada juntamente com uma descrição das funcionalidades apresentadas.

4.4 O Estudo de Caso

Os estudos de caso, buscam mostrar a utilidade do PDBest através de exemplos práticos utilizados em artigos publicados por membros do Laboratório de Bioinformática e Sistemas (LBS) e associados, ao longo de quase uma década.

Os exemplos e estudos de caso contém o direcionamento (*link*) para o arquivo de

protocolo gerado pelo PDBest, correspondente aos parâmetros usados para obter os arquivos, bem como os filtros e edições realizadas. A primeira parte do Apêndice 5.1.2 traz um exemplo com as principais funcionalidades disponíveis no PDBest e os benefícios da reprodutibilidade através do compartilhamento do arquivos de protocolo. Os três casos seguintes, tratam da utilizações práticas dos métodos que compõem o PDBest (Gonçalves et al., 2015):

- O primeiro estudo de caso, identificado como item 3.1 no Apêndice 5.1.2, foi desenvolvido por da Silveira et al. (2009) e objetivou estudar os contatos resultantes da interação entre os resíduos de uma mesma proteína (intra-cadeia). Neste estudo, o PDBest foi utilizados para criar critérios de seleção (aquisição de dados) e gerar a base de dados da pesquisa.

- No segundo estudo de caso, identificado como 3.2, Gonçalves-Almeida et al. (2012) estudou o fenômeno da inibição cruzada, por meio de ilhas hidrofóbicas na interface molecular, que permitem o reconhecimento de um mesmo inibidor por proteínas de famílias diferentes e, consequentemente, estruturas tridimensionais distintas.

- E no último estudo de caso, o objetivo foi estudar as interações não covalentes particularmente ligadas às famílias de proteínas *Cyclin-Dependent Kinase* - CDK, através de uma abordagem visual (Fassio et al., 2014).

4.5 A Publicação

A ferramenta desenvolvida foi publicada no periódico Bioinformatics de Oxford e a referência completa pode ser encontrada em Gonçalves et al. (2015).

4.6 Métricas de avaliações

É fato que nenhum dos trabalhos desenvolvidos anteriormente, descritos nesse texto, foram projetados com os mesmos objetivos do PDBest e por isso iniciativas de compará-los integralmente não são bem sucedidas, visto que seus objetivos são diferentes, embora tenham interseções. Nesta seção, para conclusão do trabalho, foi feito um comparativo do trabalho proposto com outras opções disponíveis para aquisição e tratamento de arquivos PDB. A Tabela 4.2 faz uma comparação do PDBest relacionando suas principais funcionalidades às outras soluções disponíveis.

Dada a relevância da Biologia Estrutural e Bioinformática, diversas iniciativas foram bem sucedidas no desenvolvimento de bibliotecas para a manipulação de arquivos PDB em variadas linguagens, como Python (Cock et al., 2009), Java (Holland et al., 2008), R (Grant et al., 2006) dentre outras (Gajda, 2013) amplamente usadas. Entretanto, o uso destes recursos pressupõem grande experiência com a linguagem a ser utilizada.

Tabela 4.2: Comparação de funcionalidades do PDBest e de trabalhos relacionados

Função	**Ferramenta**							
	PDBest	BIO::*	R	Pro-SA web	PDB2 PQR	PISCES	Open Babel	RCSB
Pesquisa	✓	✗	✗	✓	✓	✓	✗	✓
Download	✓	✓	✗	✓	✗	✗	✗	✓
Separação de Cadeias	✓	✓	✓	✗	✗	✗	✗	✗
Separação de Resíduos	✓	✓	✓	✗	✗	✗	✗	✗
Separação de Átomos	✓	✓	✓	✗	✗	✗	✗	✗
Separação da Cadeia Principal	✓	✓	✓	✗	✗	✗	✗	✗
Seleção de ocupância	✓	✓	✓	✗	✗	✗	✗	✗
Adição de Hidrogênio	✓	✓	✓	✗	✗	✗	✓	✗
Remoção de Hidrogênio	✓	✓	✓	✗	✗	✗	✓	✗
Renumeração de Átomos	✓	✓	✓	✗	✗	✗	✗	✗
Renumeração de Resíduos	✓	✓	✓	✗	✗	✗	✗	✗
Interface Gráfica	✓	✗	✗	✓	✓	✓	✓	✓
Reprodutibilidade	✓	✓	✓	✗	✗	✗	✗	✗
Conversão de Formatos	✓	✓	✓	✗	✗	✗	✓	✗
Não Necessita Programação	✓	✗	✗	✓	✓	✓	✓	✓

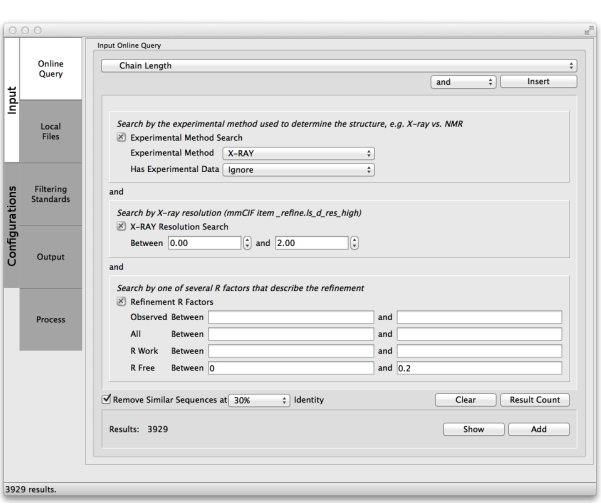
Mesmo que usuários experientes estejam dispostos a utilizar estes recursos é necessário tempo e persistência até se obter a experiência necessária para executar tarefas que são executadas de forma simples e direta no PDBest. O PDBest é justamente uma alternativa para este tipo de problema, problemas frequentes no pré-processamento de arquivos PDB por estudantes e pesquisadores são facilmente resolvidos com auxílio de uma interface gráfica interativa, pronta para uso e simples de usar.

Apesar de todas as consultas realizadas no PDBest serem submetidas ao serviço disponibilizado por Berman et al. (2000), que, comparativamente atua na pesquisa e *download* de estruturas tridimensionais, algumas características não estão presentes neste site. Um bom exemplo é a incorporação do operador lógico "OR"ao mesclar parâmetros de consulta a esta base de dados e não oferece ferramentas de otimização de qualidade estrutural. Muito embora forneça um interface amigável, para realização de consultas online e download de estruturas, a interface é limitada quando usada para baixar muitos arquivos.

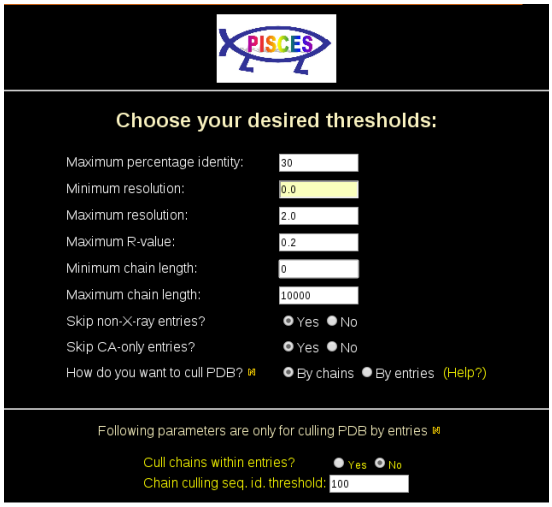
O Open Babel (OLBoyle et al., 2011) oferece opções para adição e remoção de hidrogênios e conversão de formatos que extrapolam as do PDBest entretanto, não dispõe de interface de pesquisa de estrutura e detecção de erros estruturais.

O PISCES (Wang e Dunbrack, 2005) fornece funções parecidas com o PDBest e o RCSB (Berman et al., 2000) para aquisição de arquivos. Entretanto, para obter os resultados da pesquisa é preciso cadastrar um e-mail, esperar o recebimento do resultado, que virá por e-mail e acessar um outro site externo para aquisição das estruturas. A Figura 4.4 mostra uma pesquisa similar sendo feita em cada uma das duas interfaces (PDBest e PISCES). As Figuras 4.4(a) 4.4(b) apresentam parâmetros equivalentes e a Figura 4.4(c) mostra o resultado apresentado pelo software. Notamos que há um bom atraso no recebi-

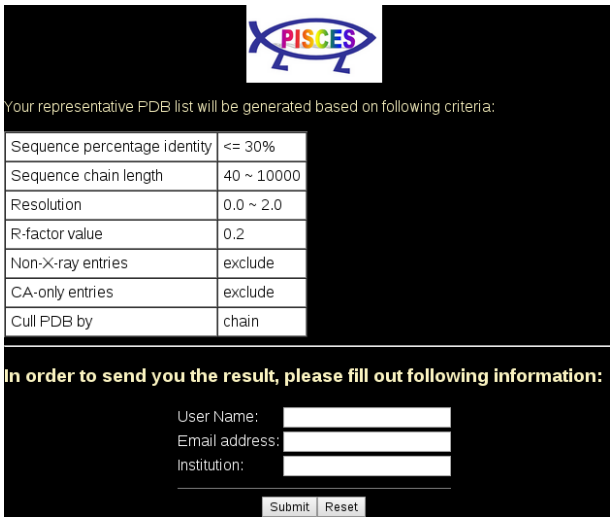
mento do e-mail com os resultados o que, a nosso ver, é uma limitação nessa ferramenta. Destacamos que com o PDBest, o usuário recebe, em tempo real, informações referentes a quantidade de arquivos que correspondem à e, após submissão da consulta, recebe prontamente a lista de identificadores e o download é feito pela própria ferramenta usando processamento paralelo o que acelera bastante o processo de aquisição de dados.



(a) Consulta PDBest



(b) Consulta PISCES



(c) Resultados PISCES

Figura 4.4: Aquisição de dados online

Fonte: http://dunbrack.fccc.edu/PISCES.php

O software ProSA-web (Wiederstein e Sippl, 2007) é um software que necessita de interação com o usuário a cada estrutura analisada e por isso não consegue aplicar um conjunto de critérios a mais de uma estrutura de forma automatizada, assim como o PDBest faz. O ProSA-web permite que consultas pontuais sejam feitas ao PDB, inserindo exatamente o identificador do arquivo ao qual se pretende utilizar. O PDBest novamente é superior a medida que oferece várias opções para eliminar ou incluir determinada estrutura.

A aplicação PDB2PQR (Dolinsky et al., 2007), oferece funções semelhantes ao PDBest, como a adição de átomos de hidrogênio e a detecção de átomos ausentes. Entretanto, não dispõe de funções de pesquisas e *downloads* para aquisição de biomoléculas, conversão de formatos e eliminação de informações dispensáveis nestes arquivos.

4.6.0.1 Tempo de processamento

O volume de dados a ser processado é um fator limitante quando consideramos as soluções atuais para manipulação de arquivos de estruturas de proteínas. Para fazer face a este problema e fornecer escalabilidade, valendo-se das arquiteturas multiprocessadas existentes, o PDBest foi implementado usando programação paralela, o que permitiu reduzir consideravelmente o tempo de processamento.

Apenas para dar uma ideia da ordem de grandeza do tempo de execução, o PDBest é capaz de filtrar o conjunto completo das estruturas depositados no banco de dados PDB, mais de 100.000 estruturas (Berman et al., 2000) em uma estação de trabalho comum utilizando oito processadores em menos de 2 horas.

4.6.0.2 Downloads

Muito embora o download de arquivos no banco de dados universal não fosse um problema, durante este projeto notou-se que ao utilizar o PDBest para fazer download de arquivos obtém-se uma significativa redução do tempo. É sabido que a medição de desempenho em redes públicas é sensível às condições exatas de tráfego no momento do experimento. Em outras palavras, a velocidade de conexão oscila muito e o acesso à serviços WEB pode ficar comprometido. Entretanto, um pequeno teste que realizou a aquisição de 7.200 estruturas para medir o tempo de início e fim da funcionalidade usando o PDBest e diretamente do RCSB (Berman et al., 2000), conforme podemos observar pela Tabela 4.3 nosso tempo é bastante reduzido pelo processo ser realizado em paralelo.

Tabela 4.3: Comparação entre os downloads efetuados pelo PDBest e RCSB

Downloads	
PDBest	6 minutos
RCSB	34 minutos

Capítulo 5

Conclusões e trabalhos futuros

Neste trabalho foi proposto o *PDB Enhance Structure Toolkit* - PDBest, que realiza operações de aquisição, manipulação, edição e conversão de estruturas tridimensionais de biomoléculas. O PDBest também permite a supervisão da aquisição das estruturas e realimentação com arquivos gerados pelo próprio software.

O PDBest foi desenvolvido usando a Linguagem de Programação C++, aliada ao poderoso *framework* multiplataforma Qt, que facilita a implementação de uma interface gráfica interativa para os principais sistemas operacionais atuais.

Ao usar programação paralela, o PDBest conseguiu, usando uma estação de trabalho (computador) comum, processar todos os dados do PDB em um tempo bastante aceitável (cerca de 2 horas).

Durante o desenvolvimento, vários testes foram realizados e a constante manutenção do software leva a crer que a versão disponibilizada no site do PDBest[1] é estável, mesmo sabendo que erros poderão ser reportados. Para tal, mantemos um canal aberto com o usuário que permite reportar problemas e sugestões através do site ou do próprio software.

Ao comparar o PDBest a outras soluções disponíveis, entendeu-se que ele foi mais bem sucedido na usabilidade e no processamento [2]. Quanto a quantidade de opções fornecidas, em relação às implementações caracterizadas como já implementadas (ferramentas prontas), o PDBest também se mostrou superior em termos da ampla abrangência de suas funcionalidades.

5.1 Trabalhos futuros

Na conclusão desse trabalho de mestrado, percebeu-se que algumas melhorias poderiam aprimorar e expandir o sistema proposto. Entre elas estão a inclusão de uma interface para receber parâmetros por linha de comando permitindo ligar o PDBest a outras apli-

[1] www.pdbest.dcc.ufmg.br

[2] O desempenho está ligado aos recursos computacionais dedicados

cações de forma automatizada e a utilização dos recursos web para implementar novas funcionalidades de visualização e utilização de recursos computacionais externos.

5.1.1 Interface por linha de comando

É inegável a necessidade do caráter interativo da utilização de uma interface gráfica para manipular as funcionalidades de um software como o PDBest de forma bastante simples e fácil, como já foi exposto nas sessões anteriores. São várias as opções que o usuário pode escolher para gerar um conjunto de dados e seu uso se dá sem grande tempo de aprendizado e treinamento. Entretanto, algumas funcionalidades, principalmente as relacionadas à edição pode ser muito útil se ligados a um outro processo automatizado, ou seja, se acopladas a sistemas desenvolvidos por outros programadores. Pretendemos a seguir avaliar essa necessidade e possivelmente projetar e implementar uma abordagem de uso do PDBest via linha de comando.

5.1.2 Interface WEB

Através da WEB é possível disponibilizar recursos computacionais e usar técnicas de visualização interativa de dados, além de facilitar o acesso já que não requer instalação prévia da aplicação, no usuário final. Planejamos desenvolver no futuro, possivelmente durante o doutorado, uma versão web do PDBest. Para que essa versão seja construída, muita reflexão e trabalho será necessária. Será quase como construir uma nova ferramenta com as mesmas funcionalidades partindo quase do zero.

Uma de nossas ideias é que essa ferramenta se transforme em uma plataforma na qual várias análise estatísticas e visuais possam resultar dos dados obtidos e processados. Gostaríamos que o usuário possa não somente pesquisar e obter os dados e tratá-los como também possa ganhar conhecimento sobre o conteúdo desses dados.Por exemplo, quantas sequências diferentes? Quantos e quais ligantes aparecem? Há proteínas mutantes? Qual o tamanho médio das sequências? Há sequências anômalas em termos de tamanho? Há grupamentos de sequências semelhantes? Qual o percentual da base que traz a estrutura completa em termos dos resíduos disponíveis? Qual o percentual da base resolvida com difração de raios X? Qual o percentual da base é proveniente de eucariotos? Essas são apenas algumas das perguntas que gostaríamos de resolver com informações ao usuário através de um pequeno relatório com as estatísticas de sua base de dados. Adicionalmente, há inúmeras outras funcionalidades que poderiam ser interessantes como o compartilhamento de bases entre usuários. Por exemplo, ao publicar um artigo, um autor poderia publicar no PDBest web sua base de dados, bem como a consulta que a gerou (para que ela possa ser atualizada gerando versões futuras) e os processamentos que foram executados na mesma. Dessa forma, outros autores poderiam facilmente obter a mesma versão da base de dados usada, bem como uma versão atualizada e com os mesmos processamentos da base original.

Referências Bibliográficas

Andreas D. Baxevanis, B. F. F. O. (2001). *Bioinformatics. A Practical Guide to the Analysis of Genes and Proteins*. Oxford University Press, 2 edição.

Ashburner, M.; Ball, C. A.; Blake, J. A.; Botstein, D.; Butler, H.; Cherry, J. M.; Davis, A. P.; Dolinski, K.; Dwight, S. S.; Eppig, J. T. et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29.

Attwood T.K., Gisel A., N.-E. E. e E., B.-R. (2011). Concepts, Historical Milestones and the Central Place of Bioinformatics in Modern Biology: A European Perspective. *Bioinformatics - Trends and Methodologies*, pp. 549–550.

Berg, J.; Tymoczko, J. e Stryer, L. (2004). Bioquímica. 5ª edição.

Berman, H.; Henrick, K. e Nakamura, H. (2003). Announcing the worldwide protein data bank. *Nature Structural & Molecular Biology*, 10(12):980–980.

Berman, H.; Henrick, K.; Nakamura, H. e Markley, J. L. (2007). The worldwide protein data bank (wwpdb): ensuring a single, uniform archive of pdb data. *Nucleic Acids Research*, 35(suppl 1):D301–D303.

Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N. e Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Research*, 28(1):235–242.

Bernstein, F. C.; Koetzle, T. F.; Willian, G. J. B.; Meyer, E. F.; Brice, M. D.; Rodgers, J. R.; Kennard, O.; Shimanouchi, T. e Tasumi, M. (1977). The protein data bank. *European Journal of Biochemistry*, 80(2):319–324.

Boutselakis, H.; Dimitropoulos, D.; Fillon, J.; Golovin, A.; Henrick, K.; Hussain, A.; Ionides, J.; John, M.; Keller, P. A.; Krissinel, E.; McNeil, P.; Naim, A.; Newman, R.; Oldfield, T.; Pineda, J.; Rachedi, A.; Copeland, J.; Sitnov, A.; Sobhany, S.; Suarez-Uruena, A.; Swaminathan, J.; Tagari, M.; Tate, J.; Tromm, S.; Velankar, S. e Vranken, W. (2003). E-msd: the european bioinformatics institute macromolecular structure database. *Nucleic Acids Research*, 31(1):458–462.

C., B. e J., T. (1999). *Introduction to Protein Structure*. Garland Science, 2 edição.

Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B. e de Hoon, M. J. L. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423.

Collaborative Computational Project, N. . (1994). The CCP4 suite: programs for protein crystallography. *Acta Crystallographica Section D*, 50(5):760–763.

Conte, L. L.; Ailey, B.; Hubbard, T. J.; Brenner, S. E.; Murzin, A. G. e Chothia, C. (2000). Scop: a structural classification of proteins database. *Nucleic acids research*, 28(1):257–259.

da Silveira, C. H.; Pires, D. E. V.; Melo-Minardi, R. C.; Ribeiro, C.; Veloso, C. J. M.; Lopes, J. C. D.; Meira Jr, W.; Neshich, G.; Ramos, C. H. I.; Habesch, R. e Santoro, M. M. (2009). Protein cutoff scanning: A comparative analysis of cutoff dependent and cutoff free methods for prospecting contacts in proteins. *Proteins: Struct., Funct., Bioinf.*, 74(3):727–743.

da Silveira, N. J. F. (2005). *BioinformÃ¡tica Estrutural Aplicada ao Estudo de ProteÃnas Alvo do Genoma do Mycobacterium tuberculosis*. PhD thesis, Universidade Estadual Paulista.

Deshpande, N.; Addess, K. J.; Bluhm, W. F.; Merino-Ott, J. C.; Townsend-Merino, W.; Zhang, Q.; Knezevich, C.; Xie, L.; Chen, L.; Feng, Z.; Kramer Green, R.; Flippen-Anderson, J. L.; Westbrook, J.; Berman, H. M. e Bourne, P. E. (2005). The rcsb protein data bank: a redesigned query system and relational database based on the mmcif schema. *Nucleic Acids Research*, 33(suppl 1):D233–D237.

Dolinsky, T. J.; Czodrowski, P.; Li, H.; Nielsen, J. E.; Jensen, J. H.; Klebe, G. e Baker, N. A. (2007). PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Res.*, 35(suppl 2):W522–W525.

Dorn, M. e Norberto de Souza, O. (2013). An interval-based algorithm to represent conformational states of experimentally determined polypeptide templates and fast prediction of approximated 3d protein structures. *International journal of bioinformatics research and applications*, 9(5):462–486.

Engh, R. A. e Huber, R. (1991). Accurate bond and angle parameters for X-ray protein structure refinement. *Acta Crystallographica Section A*, 47(4):392–400.

Fassio, A. V.; Silveira, S. A. e Melo-Minardi, R. C. (2014). Visual and interactive strategies to reveal patterns of protein-ligand interactions. In *ISCB-Latin American*.

Feldman, H. J.; Snyder, K. A.; Ticoll, A.; Pintilie, G. e Hogue, C. W. (2006). A complete small molecule dataset from the protein data bank. *{FEBS} Letters*, 580(6):1649 – 1653.

Finn, R. D.; Bateman, A.; Clements, J.; Coggill, P.; Eberhardt, R. Y.; Eddy, S. R.; Heger, A.; Hetherington, K.; Holm, L.; Mistry, J.; Sonnhammer, E. L. L.; Tate, J. e Punta, M. (2014). Pfam: the protein families database. *Nucleic Acids Research*, 42(D1):D222–D230.

Fourment, M. e Gillings, M. R. (2008). A comparison of common programming languages used in bioinformatics. *BMC bioinformatics*, 9(1):82.

Gajda, M. J. (2013). hPDB-Haskell library for processing atomic biomolecular structures in Protein Data Bank format. *BMC Res. Notes*, 6(1):483.

Gelbin, A.; Schneider, B.; Clowney, L.; Hsieh, S.-H.; Olson, W. K. e Berman, H. M. (1996). Geometric parameters in nucleic acids: Sugar and phosphate constituents. *Journal of the American Chemical Society*, 118(3):519–529.

Golovin, A.; Oldfield, T.; Tate, J. G.; Velankar, S.; Barton, G. J.; Boutselakis, H.; Dimitropoulos, D.; Fillon, J.; Hussain, A.; Ionides, J. M. et al. (2004). E-msd: an integrated data resource for bioinformatics. *Nucleic Acids Research*, 32(suppl 1):D211–D216.

Gonçalves, W. R. S.; Gonçalves-Almeida, V. M.; Arruda, A. L.; Meira, W.; da Silveira, C. H.; Pires, D. E. V. e de Melo-Minardi, R. C. (2014). Pdbest: a user-friendly platform for manipulating and enhancing protein structures.

Gonçalves, W. R. S.; Gonçalves-Almeida, V. M.; Arruda, A. L.; Meira, W.; da Silveira, C. H.; Pires, D. E. V. e de Melo-Minardi, R. C. (2015). Pdbest: a user -friendly platform for manipulating and enhancing protein structures. *Bioinformatics*.

Gonçalves-Almeida, V. M.; Pires, D. E. V.; de Melo-Minardi, R. C.; da Silveira, C. H.; Meira, W. e Santoro, M. M. (2012). Hydropace: understanding and predicting cross-inhibition in serine proteases through hydrophobic patch centroids. *Bioinformatics*, 28(3):342–349.

Grant, B. J.; Rodrigues, A. P.; ElSawy, K. M.; McCammon, J. A. e Caves, L. S. (2006). Bio3D: an R package for the comparative analysis of protein structures. *Bioinformatics*, 22(21):2695–2696.

Henrick K., Berman H.M., N. H. (2005). The protein data bank and the wwpdb. *Encyclopedia of Genomics, Proteomics, and Bioinformatics*, 7:3335–3339.

Holland, R. C.; Down, T. A.; Pocock, M.; Prlić, A.; Huen, D.; James, K.; Foisy, S.; Dräger, A.; Yates, A.; Heuer, M. et al. (2008). Biojava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097.

Lesk, A. M. (2001). *Introduction to Protein Architecture: The Structural Biology of Proteins*. Oxford University Press, 1 edição.

Lesk, A. M. (2014). *Introduction to Bioinformatics*. Oxford University Press, 4 edição.

Luscombe, N.; Greenbaum, D. e Gerstein, M. (2001). Review: What is bioinformatics? an introduction and overview. *Yearbook of Medical Informatics*, pp. 83–99.

Nelson, D. L. e Cox, M. M. (2011). *Princípios de bioquímica de Lehninger*. Artmed, 5 edição.

Nissink, J. W. M.; Murray, C.; Hartshorn, M.; Verdonk, M. L.; Cole, J. C. e Taylor, R. (2002). A new test set for validating predictions of protein-ligand interaction. *Proteins: Structure, Function, and Bioinformatics*, 49(4):457–471.

OLBoyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T. e Hutchison, G. R. (2011). Open babel: An open chemical toolbox. *J Cheminf*, 3:33.

Pires, D.; Silveira, C.; Santoro, M. e Meira Jr, W. (2007). Pdbest pdb enhanced structures toolkit. *Proceedings of the 3rd International Conference of Brazil Association for Bioinformatics*, p. 39.

Rezende, D. A. (2005). *Engenharia de software e sistemas de informação*. Brasport.

Rich, A.; Sarma, R. e Sundaralingam, M. (1983). Abbreviations and symbols for the description of conformations of polynucleotide chains. *European Journal of Biochemistry*, 131(1):9–15.

Schierz, A. C.; Soldatova; N., L. e King, R. D. (2007). Overhauling the pdb. *Nat Biotech*, 25(4):442.

Stajich, J. E.; Block, D.; Boulez, K.; Brenner, S. E.; Chervitz, S. A.; Dagdigian, C.; Fuellen, G.; Gilbert, J. G.; Korf, I.; Lapp, H. et al. (2002). The bioperl toolkit: Perl modules for the life sciences. *Genome research*, 12(10):1611–1618.

Summerfield, M. (2010). *Advanced Qt Programming: Creating Great Software with C++ and Qt 4*. Prentice Hall Press.

Szabadka, Z. e Grolmusz, V. (2007). High throughput processing of the structural information in the protein data bank. *Journal of Molecular Graphics and Modelling*, 25(6):831 – 836.

Tagari, M.; Tate, J.; Swaminathan, G.; Newman, R.; Naim, A.; Vranken, W.; Kapopoulou, A.; Hussain, A.; Fillon, J.; Henrick, K. et al. (2006). E-msd: improving data deposition and structure quality. *Nucleic acids research*, 34(suppl 1):D287–D290.

Tanenbaum, A. S. e Woodhull, A. S. (2008). *Sistemas Operacionais: Projetjos e Implementação*. Bookman.

Ulrich, E.; Markley, J. e Kyogoku, Y. (1988). Creation of a nuclear magnetic resonance data repository and literature database. *Protein sequences & data analysis*, 2(1):23–37.

Verli, H. (2014). *BioinformÃ¡tica da Biologia Ã flexibilidade molecular*. UFRGS, 1 edição.

Voet, D.; Voet, J. G. e Pratt, C. W. (2014). *Fundamentos de Bioquímica-: A Vida em Nível Molecular*. Artmed Editora.

Wako, H.; Kato, M. e Endo, S. (2004). Promode: a database of normal mode analyses on protein molecules with a full-atom model. *Bioinformatics*, 20(13):2035–2043.

Wang, G. e Dunbrack, R. L. (2005). PISCES: recent improvements to a PDB sequence culling server. *Nucleic Acids Res.*, 33(suppl 2):W94–W98.

Westbrook, J.; Feng, Z.; Burkhardt, K. e Berman, H. M. (2003). Validation of protein structures for protein data bank. In Charles W. Carter, J. e Sweet, R. M., editores, *Macromolecular Crystallography, Part D*, volume 374 of *Methods in Enzymology*, pp. 370 – 385. Academic Press.

Wiederstein, M. e Sippl, M. J. (2007). ProSA-web: interactive web service for the recognition of errors in three-dimensional structures of proteins. *Nucleic Acids Res.*, 35(suppl 2):W407–W410.

Publicação

Bioinformatics Advance Access published May 11, 2015

Bioinformatics, 2015, 1–3
doi: 10.1093/bioinformatics/btv223
Advance Access Publication Date: 24 April 2015
Applications note

Structural bioinformatics

PDBest: a user–friendly platform for manipulating and enhancing protein structures

Wellisson R. S. Gonçalves[1,*], Valdete M. Gonçalves-Almeida[1], Aleksander L. Arruda[1], Wagner Meira Jr.[1], Carlos H. da Silveira[2], Douglas E. V. Pires[3,*†], Raquel C. de Melo-Minardi[1,*†]

[1]Department of Computer Science, Universidade Federal de Minas Gerais, Brazil, [2]Advanced Campus at Itabira, Universidade Federal de Itajubá, Brazil and [3]Centro de Pesquisas René Rachou, Fundação Oswaldo Cruz, Brazil

*To whom correspondence should be addressed.

Associate Editor: Anna Tramontano

†The authors wish it to be known that, in their opinion, the last 2 authors should be regarded as Joint First Authors

Received on March 6, 2015; revised on April 10, 2015; accepted on April 19, 2015

Abstract

Summary: PDBest (PDB Enhanced Structures Toolkit) is a user-friendly, freely available platform for acquiring, manipulating and normalizing protein structures in a high-throughput and seamless fashion. With an intuitive graphical interface it allows users with no programming background to download and manipulate their files. The platform also exports protocols, enabling users to easily share PDB searching and filtering criteria, enhancing analysis reproducibility.
Availability and implementation: PDBest installation packages are freely available for several platforms at http://www.pdbest.dcc.ufmg.br
Contact: wellisson@dcc.ufmg.br, dpires@dcc.ufmg.br, raquelcm@dcc.ufmg.br
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Acquiring, manipulating and filtering large sets of biomolecule files from repositories such as the Protein Data Bank, is a common and important preprocessing step prior to protein structure studies. By treating these files one aims to improve data quality, identifying and/or removing common idiosyncrasies that could compromise and bias subsequent analysis. Common filtering tasks include, but are not limited to: filtering multiple occupancies and models, splitting files by chain, selecting subsets of atoms or residues, identifying missing atoms or residues, generating an assessment report.

Given the relevance of Structural Biology and Bioinformatics, several initiatives have been successful in developing software tools and libraries for manipulating PDB files for different programming languages, such as Python (Cock *et al.*, 2009), Java (Holland *et al.*, 2008), R (Grant *et al.*, 2006) and others (Gajda, 2013), which are widely used today.

It is, however, a task of great relevance for both educational and scientific purposes to make the manipulation and study of such source of data more accessible to a larger public (from undergraduate students to researchers) without requiring a programming background, while taking into account the ever increasing scale of current data availability and guaranteeing the reproducibility of the process.

Despite the efforts of the community to make these capabilities available as web services (Dolinsky *et al.*, 2007; Hussain *et al.*, 2002; Wang *et al.*, 2005; Wiederstein and Sippl, 2007), there still is a significant demand for an easy-to-use, scalable platform for treating protein structures.

To fill this gap, we have developed PDBest (PDB Enhanced Structures Toolkit), a user-friendly, freely available platform for acquiring, manipulating and normalizing protein structures in a reproducible, high-throughput and seamless fashion.

2 Platform description and workflow

PDBest was developed aiming to achieve four major goals: (i) provide means to improve data quality in Structural Biology/Bioinformatics; (ii) make it accessible for a larger public; (iii)

provide scalability and parallel processing; (iv) as well as reproducibility and reusability.

To establish a high-quality structural database, PDB files of interest need to be collected and properly preprocessed and/or cleaned. Many of these cleaning tasks are common to different studies (as the example in Fig. S1 of Supplementary Material), including treating NMR models or multiple occupancies while others are more specific (e.g. filtering subsets of residues or atoms).

To cope with these demands PDBest provides an intuitive graphical user interface (GUI), as depicted on Figure 1A. It was designed to make these tasks accessible to a broader public since no command line or programming experience is require, and also taking advantage of the emergence of multi-core architectures, dramatically reducing processing time via parallel processing. The software platform was developed in C++ language on the QT framework, providing high performance for all major operating systems: Windows, Unix/Linux and Mac OS X. Many capabilities, including adding hydrogens and file format conversions, were implemented using the OpenBabel library (OLBoyle *et al.*, 2011).

To achieve reproducibility, we have developed the PDBest protocol file. The protocol file encompasses all user-defined filtering preferences, data collection and filtering options that can be easily shared, improving work reproducibility and the creation of filtering standards. Figure 1B shows PDBest capabilities and pipeline which is divided in the stages described next.

Data Acquisition: Structure files can be loaded into PDBest in three ways: (i) by submitting an 'Online Query' to the RCSB PDB mirror; (ii) by loading local files or (iii) a combination of both. Using (i) users are able to search for structures using all parameters available at the RSCB web site, performing complex queries including logical operators 'and'/'or' in any combination. No restriction on the number of PDB files to be acquired is imposed. Online queries are also compatible with those of well established servers, like the PISCES web server (Wang *et al.*, 2005). Figure 1A shows how a PISCES-like query can be performed using PDBest.

Filtering Standards: After acquiring the PDB files users can set preprocessing steps to be applied. It is possible to edit or filter any information described in the PDB File Format Documentation, as well as converting between known biological file formats. Amongst the main filtering capabilities are included: splitting files by chain, selecting models, parsing ligands, adding/removing hydrogens, treating multiple occupancies, renumbering residues and atoms as well as selecting subsets of atoms or residues of interest.

Output and Processing: Processed PDB files are generated, with new suffixes defined by the user. A detailed report can be presented highlighting files with identifies issues (missing residues, missing atoms, multiple occupancies or any inconsistencies).

Reproducibility with PDBest—Protocol file: PDBest can also generate a protocol file which stores user section information, filtering parameters, online query and steps employed to generate the data set. This enables users to reuse and share data acquisition and preparation steps, improving reproducibility of the work and subsidizing the creation of processing standards. A tutorial with examples of usage is available as Supplementary Material.

3 Conclusion

PDBest offers a powerful, user-friendly interface between researchers/students in Structural Biology and common processing tasks for protein structural files. Its graphical interface provides an efficient, yet accessible, way to acquire, manipulate and check PDB files as means to improve data quality and integrity.

The platform is capable of parallel processing, taking advantage of current multi-core architectures and providing scalability for its pipeline. On top of that, PDBest enables users to share acquisition and filtering protocols, improving work reproducibility and the creation of filtering standards for research groups.

PDBest is available for all major Operating Systems and has been thoroughly tested and proved fundamental on many structure-based studies in our group including: inter-residue contact analysis (da Silveira *et al.*, 2009), receptor-based ligand prediction (Pires *et al.*, 2013),

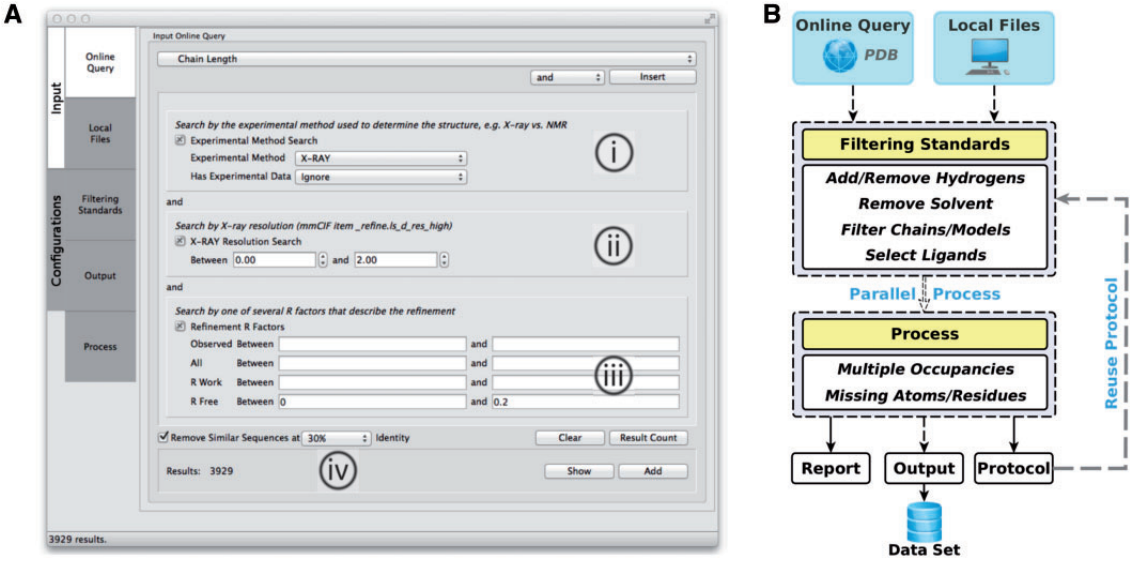


Fig. 1. PDBest GUI and workflow. (**A**) depicts the platform's main GUI features and shows the resulting screen from a PISCES-like (Wang *et al.*, 2005) online query using PDBest. The filters considered in this example were: (i) proteins solved via X-ray crystalography; (ii) X-ray resolution ≤ 2 Å; (iii) R-Free ≤ 0.2 and (iv) mutual sequence similarity lower than 30%. These filters resulted in 3929 PDB IDs, as of April 2015. The workflow in (**B**) highlights the platform's main acquisition and filtering capabilities

mutation analysis (Pires *et al.*, 2014) as well as contact network analysis (Gonçalves-Almeida *et al.*, 2012).

Funding

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES); Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); Centro de Pesquisas René Rachou (CPqRR - FIOCRUZ Minas); Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG); Financiadora de Estudos e Projetos (FINEP) and Pró-Reitoria de Pesquisa da UFMG.

Conflict of Interest: none declared.

References

Cock,P.J. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.

da Silveira,C.H. *et al.* (2009) Protein cutoff scanning: a comparative analysis of cutoff dependent and cutoff free methods for prospecting contacts in proteins. *Proteins Struct. Funct. Bioinf.*, **74**, 727–743.

Dolinsky,T.J. *et al.* (2007) PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Res.*, **35**, W522–W525.

Gajda,M.J. (2013) hPDB-Haskell library for processing atomic biomolecular structures in Protein Data Bank format. *BMC Res. Notes,* **6**, 483.

Gonçalves-Almeida,V.M. *et al.* (2012) HydroPaCe: understanding and predicting cross-inhibition in serine proteases through hydrophobic patch centroids. *Bioinformatics,* **28**, 342–349.

Grant,B.J. *et al.* (2006) Bio3D: an R package for the comparative analysis of protein structures. *Bioinformatics,* **22**, 2695–2696.

Holland,R.C. *et al.* (2008) BioJava: an open-source framework for bioinformatics. *Bioinformatics,* **24**, 2096–2097.

Hussain,A. *et al.* (2002) PDB Goodies-a web-based GUI to manipulate the Protein Data Bank file. *Acta Crystallogr. Sect. D,* **58**, 1385–1386.

OLBoyle,N.M. *et al.* (2011) Open Babel: An open chemical toolbox. *J. Cheminform.*, **3**, 33.

Pires,D.E.V. *et al.* (2013) aCSM: noise-free graph-based signatures to large-scale receptor-based ligand prediction. *Bioinformatics,* **29**, 855–861.

Pires,D.E.V. *et al.* (2014) mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics,* **30**, 335–342.

Wang,G. and Dunbrack,R.L. (2005) PISCES: recent improvements to a PDB sequence culling server. *Nucleic Acids Res.*, **33**, W94–W98.

Wiederstein,M. and Sippl,M.J. (2007) ProSA-web: interactive web service for the recognition of errors in three-dimensional structures of proteins. *Nucleic Acids Res.*, **35**, W407–W410.

Manual

PDBest Help Center

What is PDBest ?

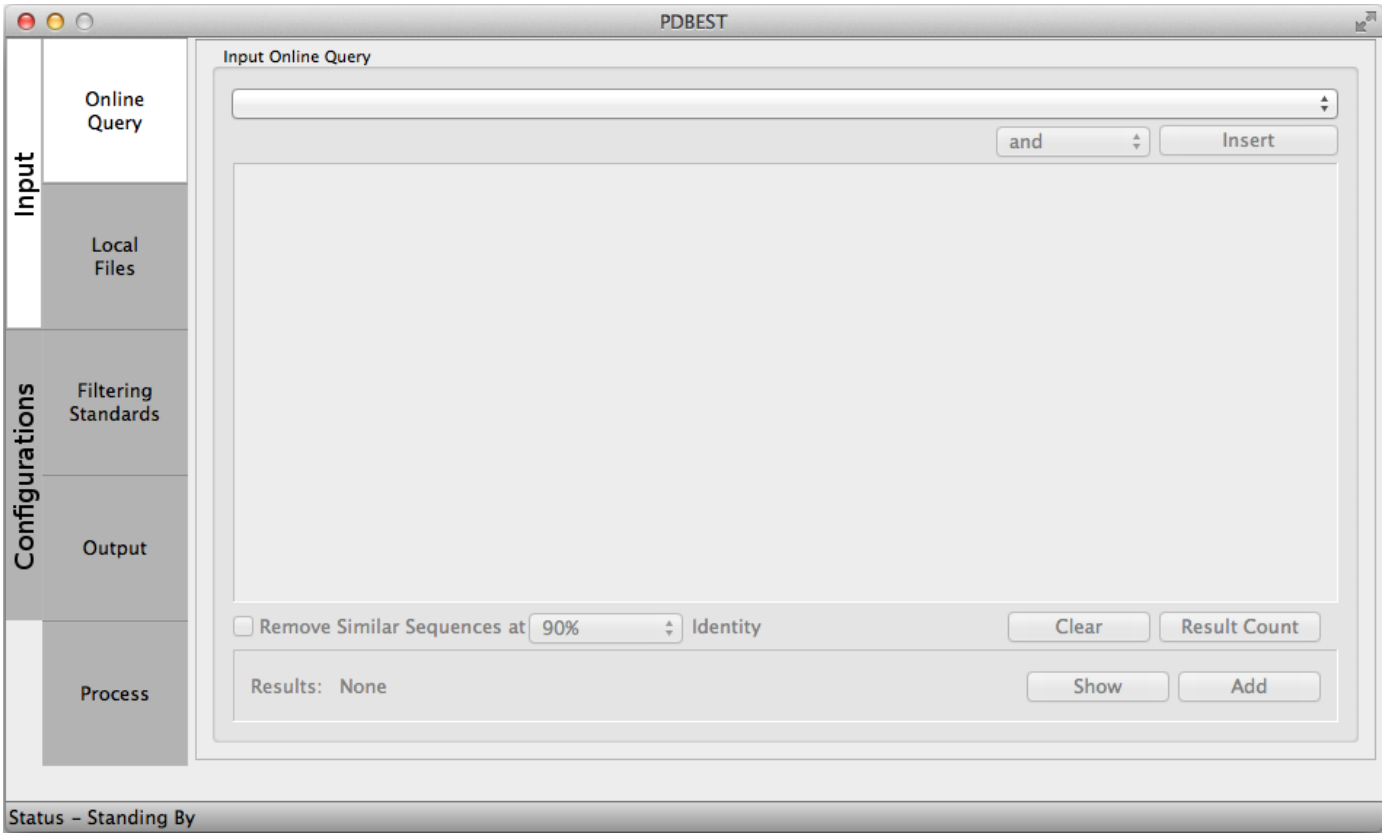
PDBest (PDB Enhanced Structures Toolkit) is a user- friendly, freely available platform for acquiring, manipulating and normalizing protein structures in a high-throughput and seamless fashion. The platform has an intuitive graphical interface developed to allow researchers and students with no programming background to download and manipulate theirs files without using the command line. The platform can also save protocols, enabling users to easily share PDB searching and filtering data, improving reproducibility of the analyses carried out subsequently.

The software platform was developed in C++ language on the QT framework, providing high performance for all major operating systems: Windows, Linux, and Mac OS X.

Input Files

On PDBest, users can provide input files from (1) a local repository, (2) download them from the RCSB Protein Data Bank (http://pdb.org) mirror, via an online query, using their searching parameters or (3) a combination of both.

OnlineQuery



Using the "Online Query" option users can search for biomolecules using all parameters available at the RCSB Protein Data Bank (http://pdb.org), combining the searching criteria with the logical operators "and"/"or" in any combination, allowing very specific and sophisticated queries to be performed.

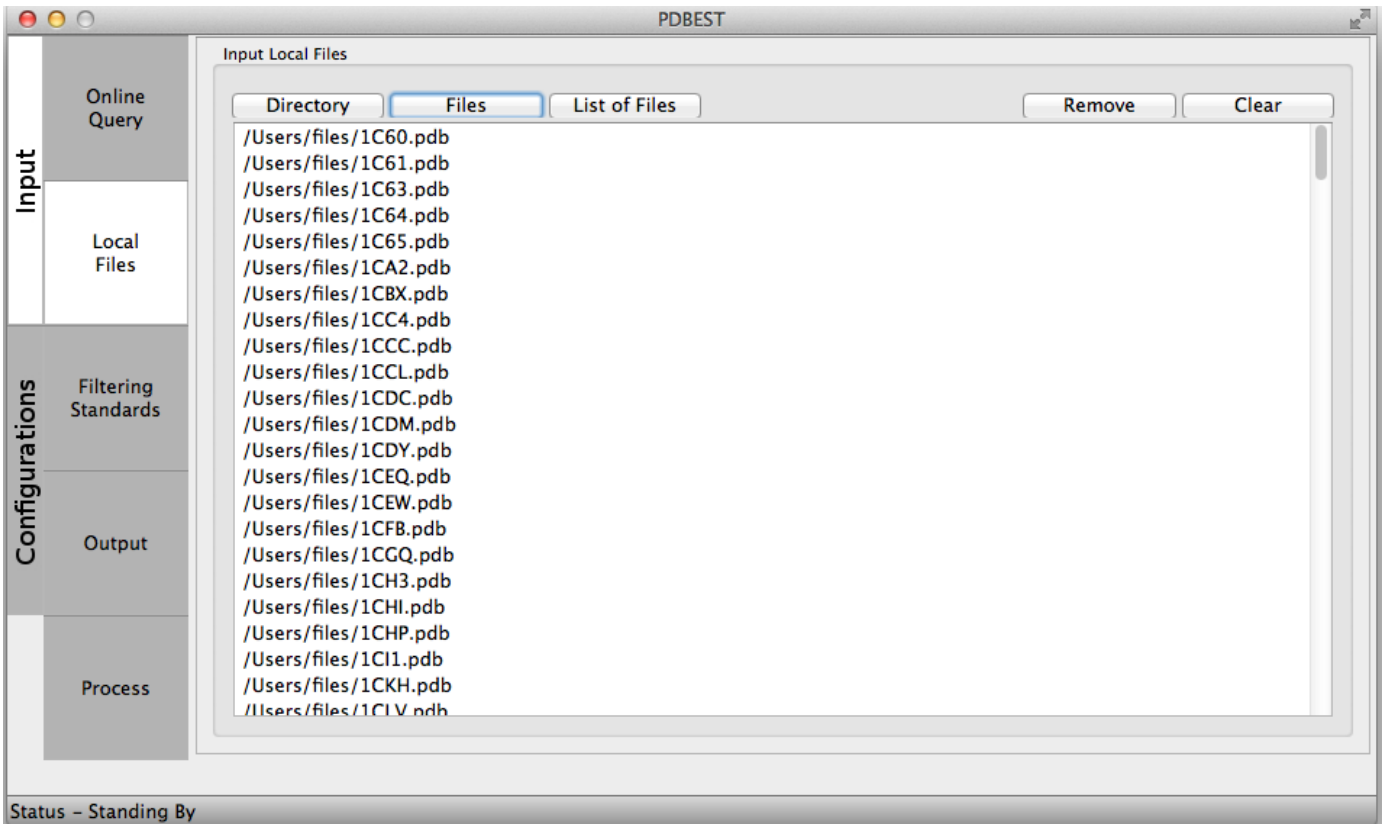
Furthermore, it is possible to remove files by different sequence similarity thresholds on **"Remove Similar Sequence at"** list box.

Users can compose their queries via the box menu and add it with the **"Insert"** button and add or modify its parameters. A query option can be removed using the icon "x" on the window or the the query can be reset with the **"clear"** button. The **"Submit"** button will submit the query to the RCSB database and the list of matching PDB files will be shown to be analyses before the processing step. It is possible to change or refine the query before processing, making adjustments at any time. The PDB identifiers can be seen via the **"Show"** button and a list of them can be saved.

The **"Add"** button can be used to manually include PDB identifiers.

There is no limit of number of molecues to be acquired and processed by PDBest.

Local Files



The input PDB files can provided from a local repository though three option buttons:

- **Files:** PDB files are selected manually from a folder.
- **Directory:** Include files selecting a whole directory, which will look for files with PDB (.pdb) or mmCIF (.cif) extentions. It is important notice that this option does not include files on sub-directories.
- **List of files:** Providing a file with a list of files to be included (the full paths must be provided).

PDB files must have unique names, even if in different directories, otherwise only one instance will be considered. Files shown on the list box are sent to processing section.

Configurations

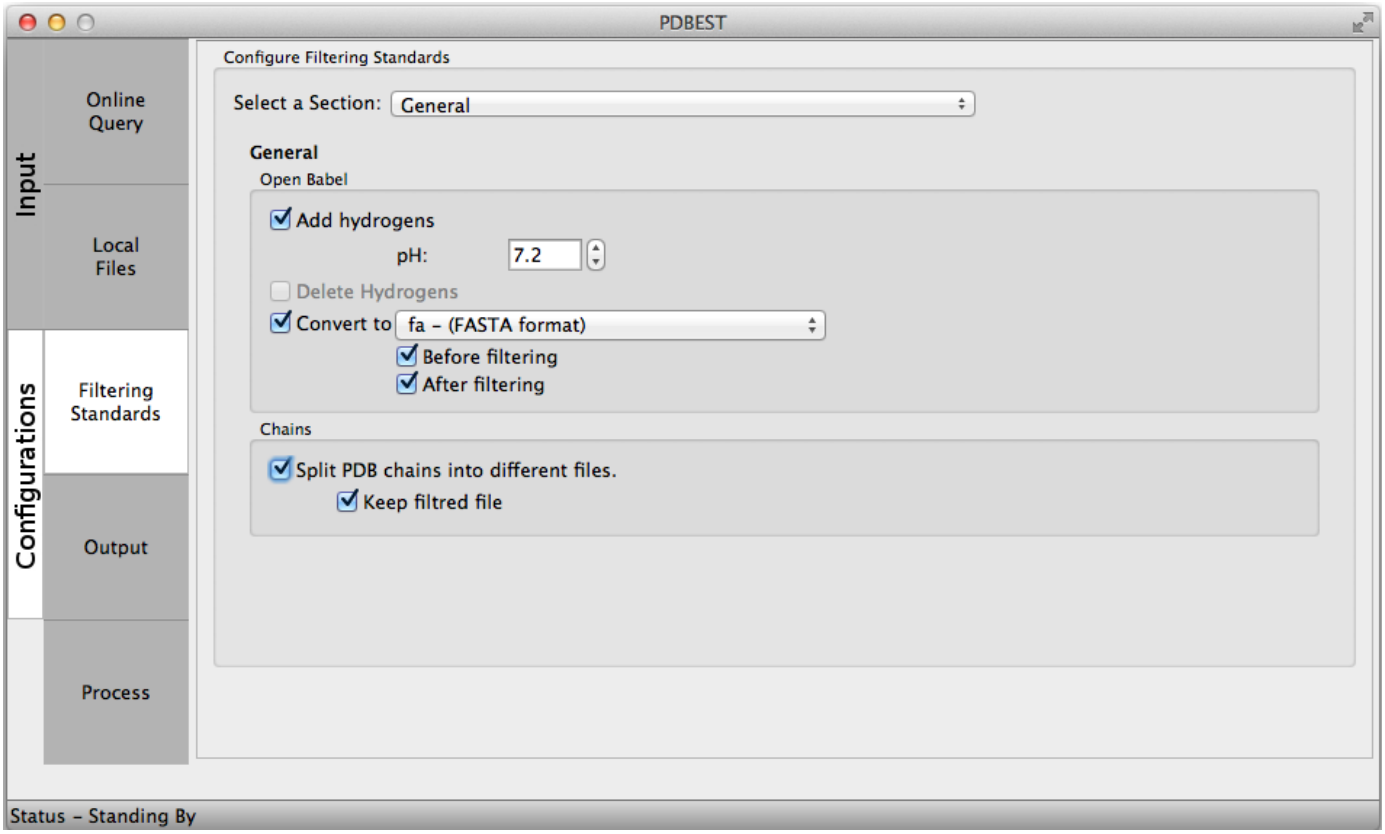
The "Configurations" section includes two main options, **"Filtering Standards"** and **"Output"** options. After loading the files users can choose amongst many filtering options to be applied as well as decide where to store the filtered files and name conventions.

Filtering Standards

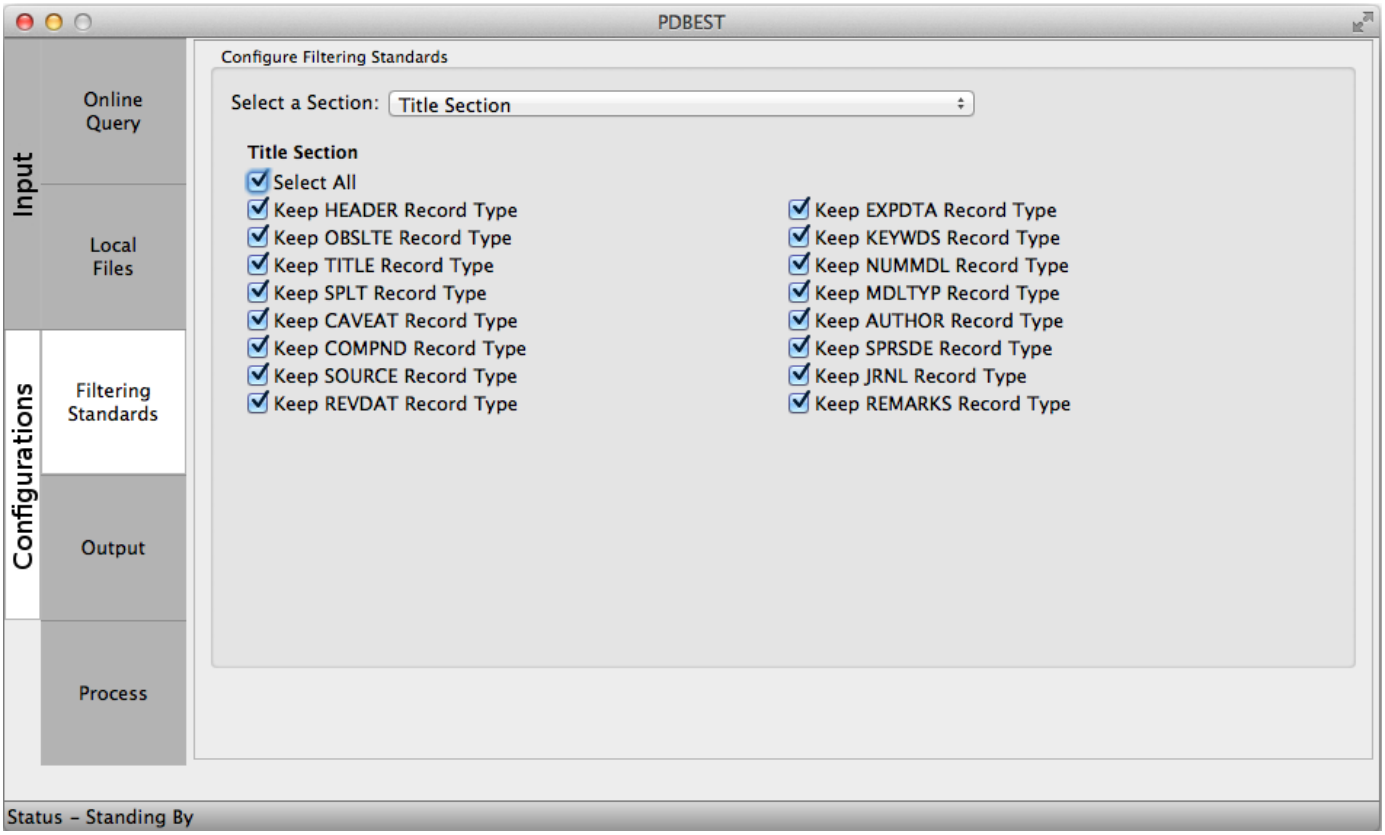
PDBest can manipulate PDB files by applying a series of filters or processing parameters which allows users to select relevant information to theirs analyses. A new file will be created with the selected records, and the original file will be maintained.

The PDB file format definition stablishes a set of standards to be followed during structure deposition and its sections are extensively described at the wwwPDB (http://www.wwpdb.org/docs.html). The PDBest filterig criteria is devided in the folloing sections. The user can choose to keep or discard the records selecting the check box accordingly:

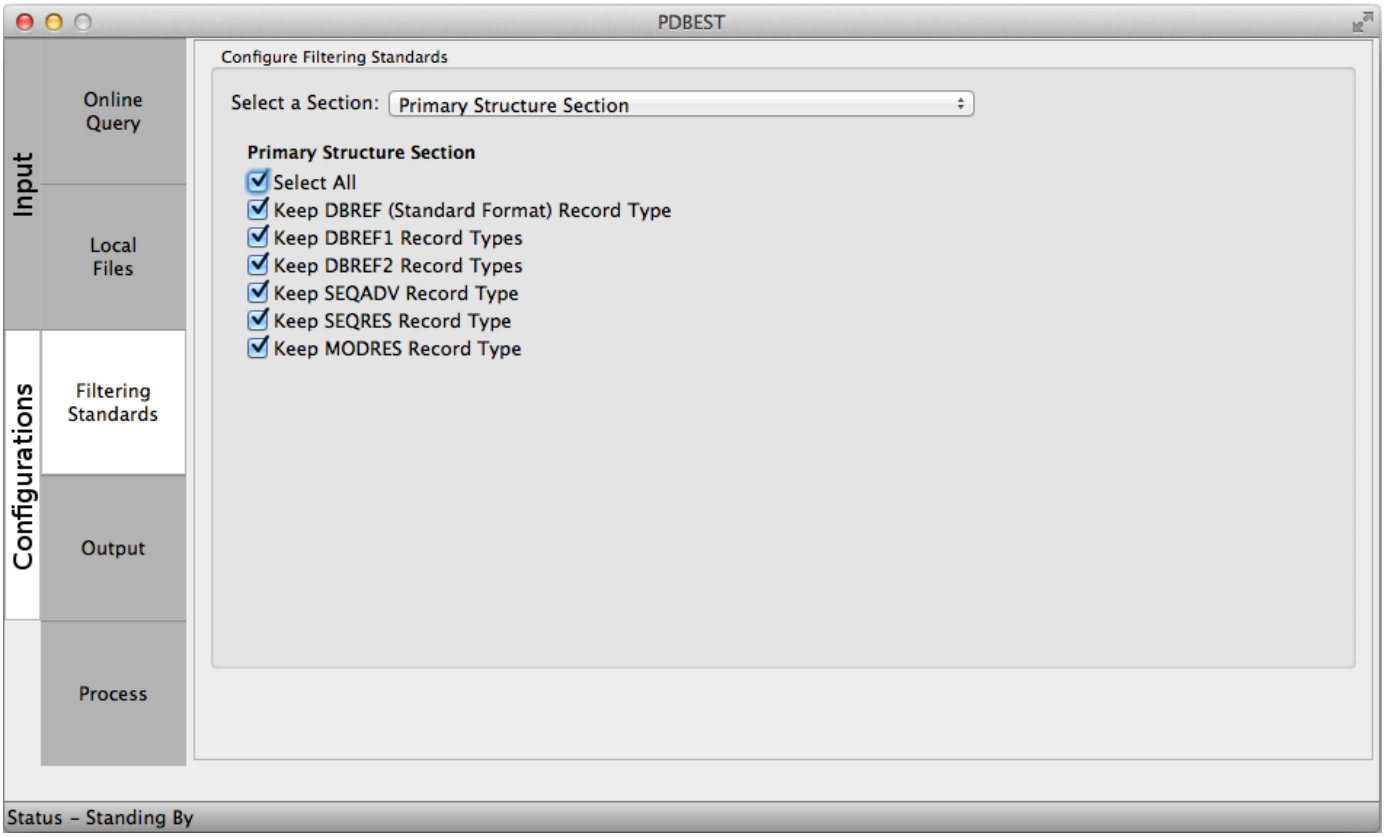
- **General:** The general section allows file format conversions, to add (or to remove) hydrogens at a given pH, split files by chain amongst other common tasks.



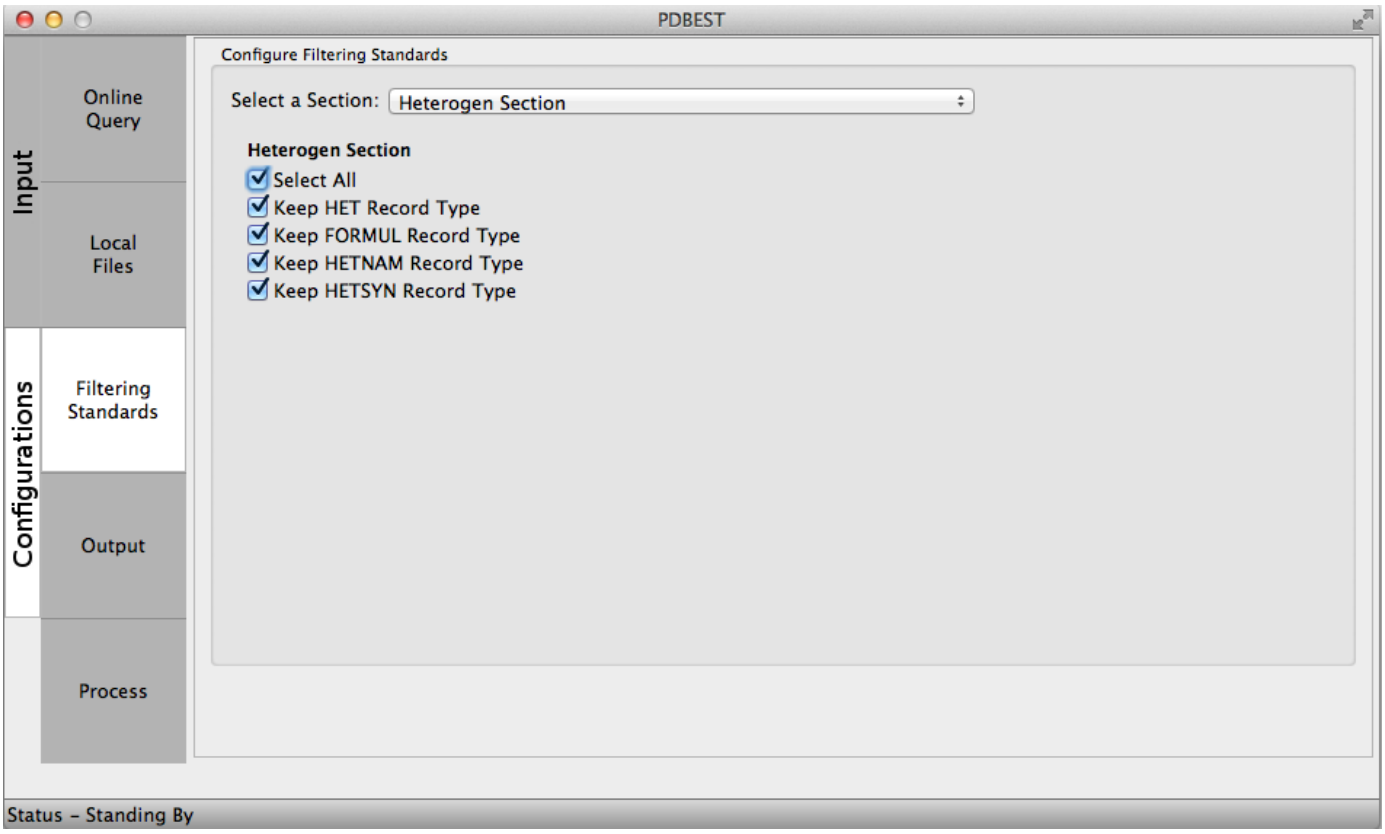
- **Title:** The title section processes records used to describe the experimental considtions and the biological macromolecules present in the entry. It includes the records: HEADER, OBSLTE, TITLE, CAVEAT, COMPND, SOURCE, KEYWDS, EXPDTA, AUTHOR, REVDAT, SPRSDE, JRNL, and REMARK.



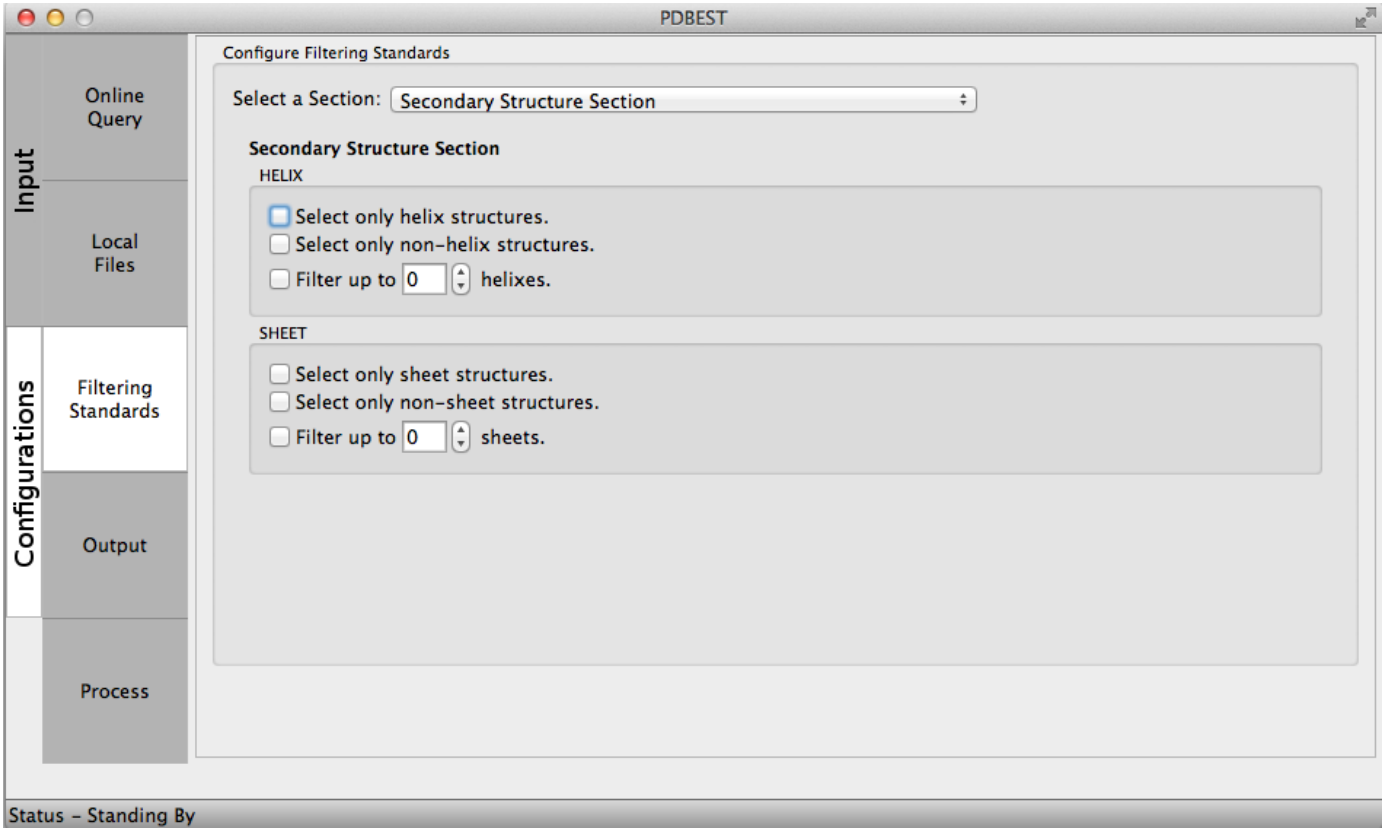
- **Primary Structure :** The primary structure section of a PDB file contains the sequence of residues in each chain of the macromolecule. The records used to define the sequence of residues are DBREF, DBREF1, DBREF2, SEQADV, SEQRES and MODRES.



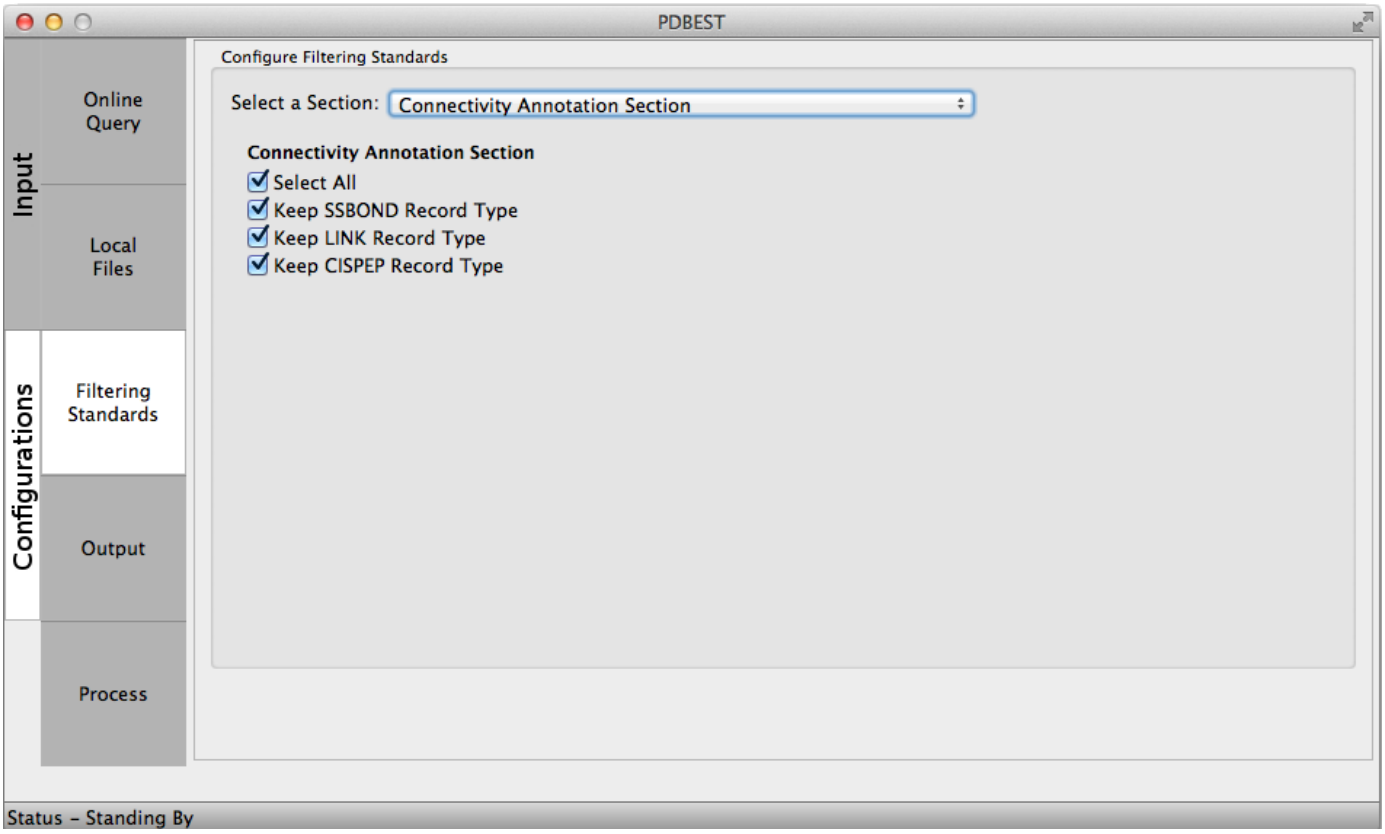
- **Heterogen :** The heterogen section of a PDB file contains the complete description of non-standard residues in the entry. The records associated to non-standard residues are HET, HETNAM, HETSYN and FORMUL.



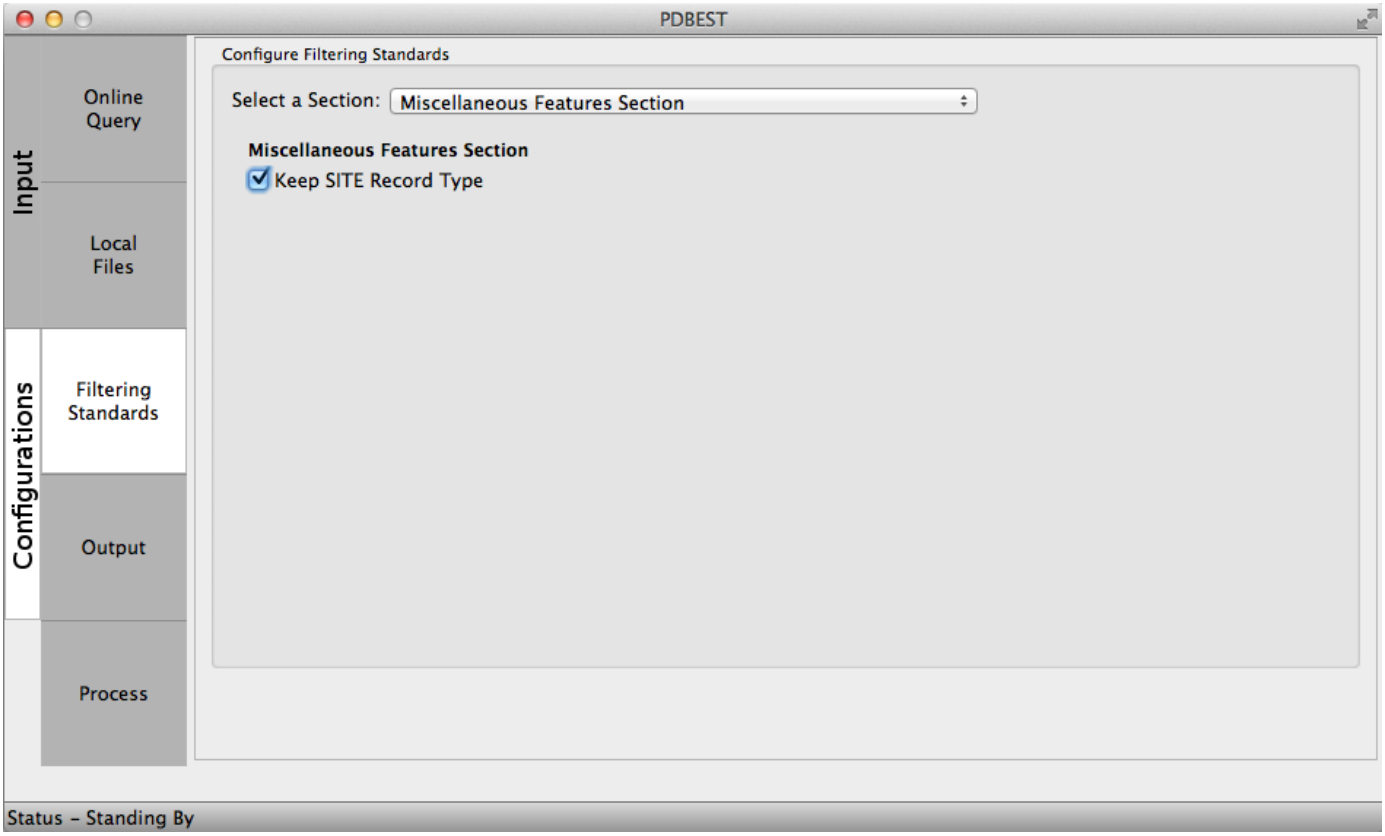
- **Secondary Structure :** The secondary structure section of a PDB file describes helices, sheets, and turns found in the protein or polypeptide. The records that describe the secondary structures are HELIX, TURN and SHEET. The user can filter residues based on its presence (or absence) on these secondary structures .



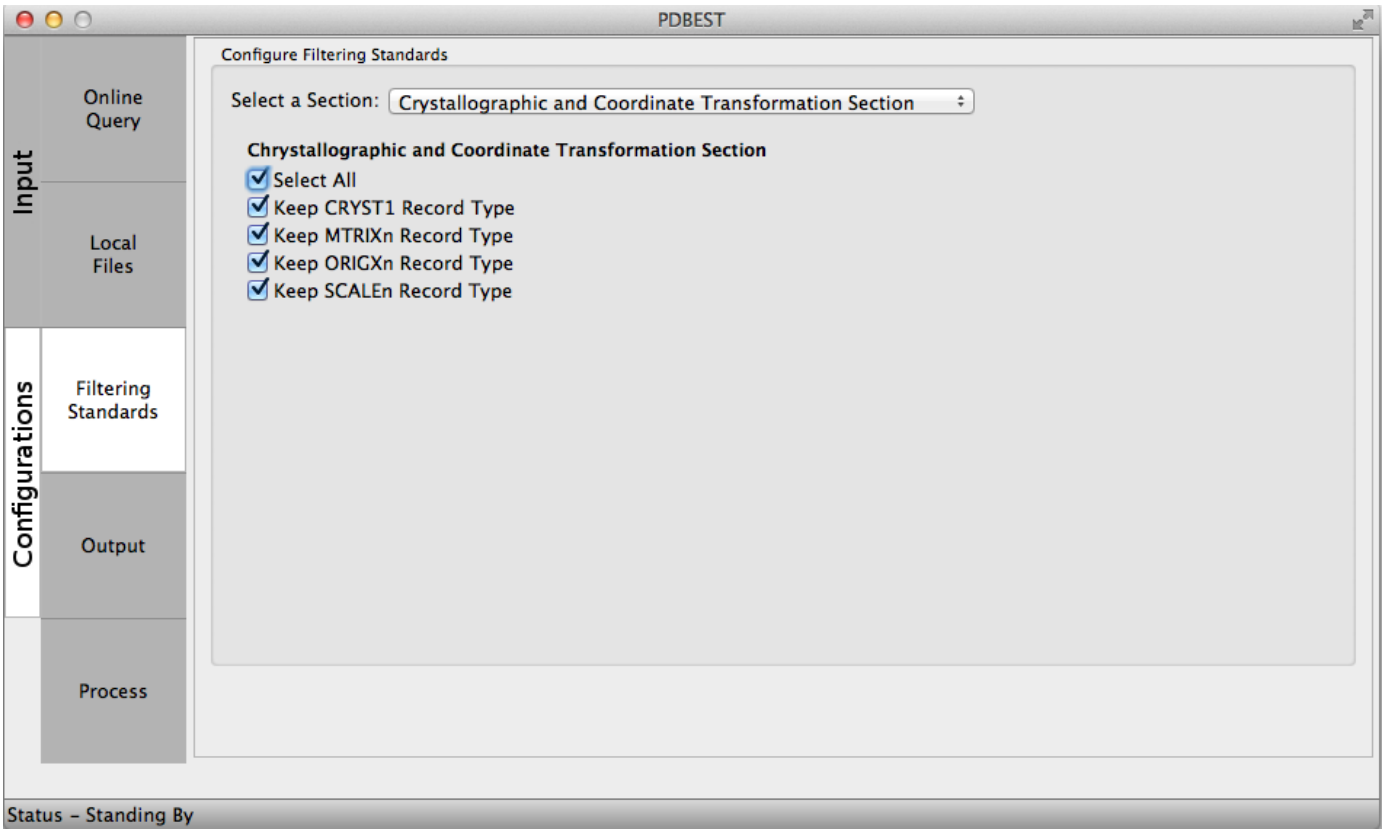
- **Connectivity Annotation :** The connectivity annotation section specifies the existence and location of disulfide bonds and other linkages. The records defining connectivity annotation are SSBOND, LINK and CISPEP. It is necessary to select the check box to keep this record.



- **Miscellaneous Features :** The miscellaneous features section describes features in the molecule such as environments surrounding a non-standard residue or an active site.



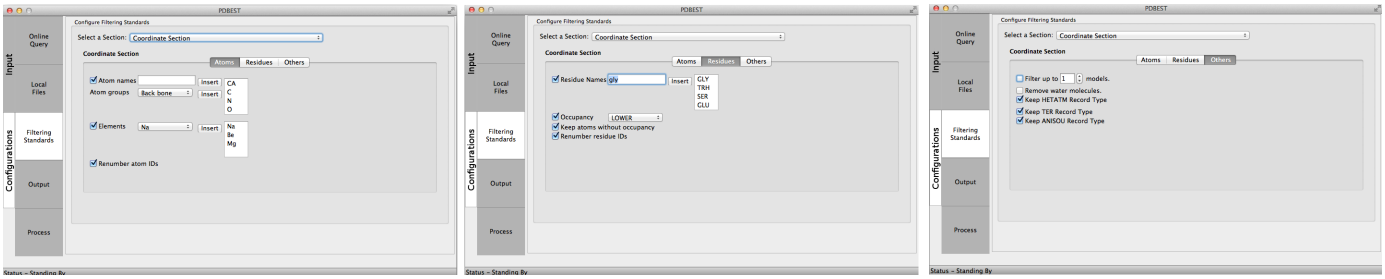
- **Crystallographic and Coordinate Transformation :** The crystallographic section describes the geometry of the crystallographic experiment and the coordinate system transformations. The records that contains these informations are CRYST1, ORIGXn, SCALEn, MTRIXn and TVECT.



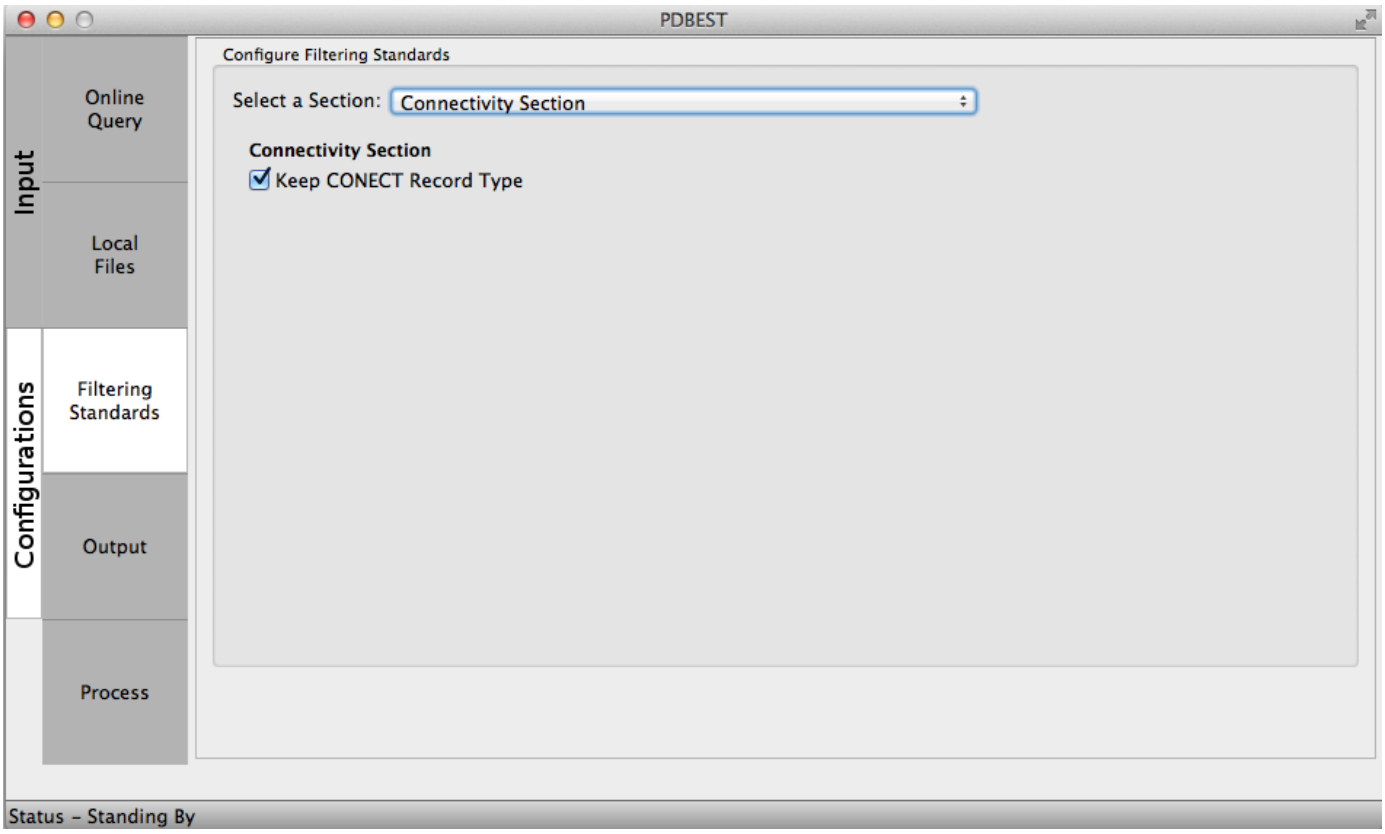
- **Coordinate :** The coordinate section contains the collection of atomic coordinates as well as the MODEL and ENDMDL records. The GUI has three tabs, one for each atomi component (atoms, residues and others). On the atom component tab users can choose filtering options by atom name (e.g., filtering backbone atoms - CA, C, N, O), and select renumbering options.

On the residue component tab users can choose to select or remove specific residues based on their 3-letter code. It is also possible to filter atoms with multiple occipancies (greater/lower occupancy or keep atoms without occupancy). It is also possible to renumber residues.

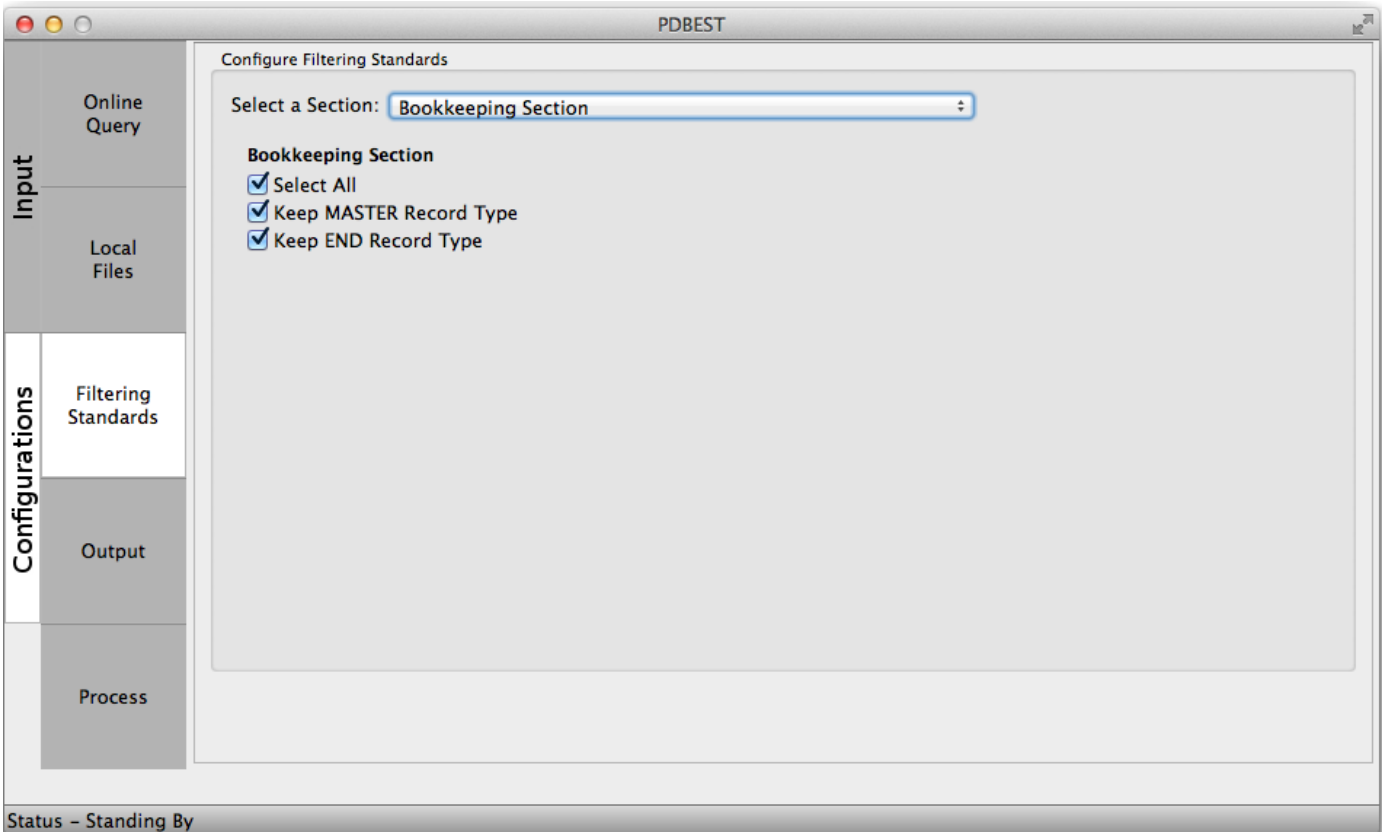
On the "other informations" tab it is possible to filter models, for instance, from structures determined by NMR, to remove solvent molecules and filter TER, ANISOU and HETATM records.



- **Connectivity :** The connectivity section provides information on chemical connectivity.

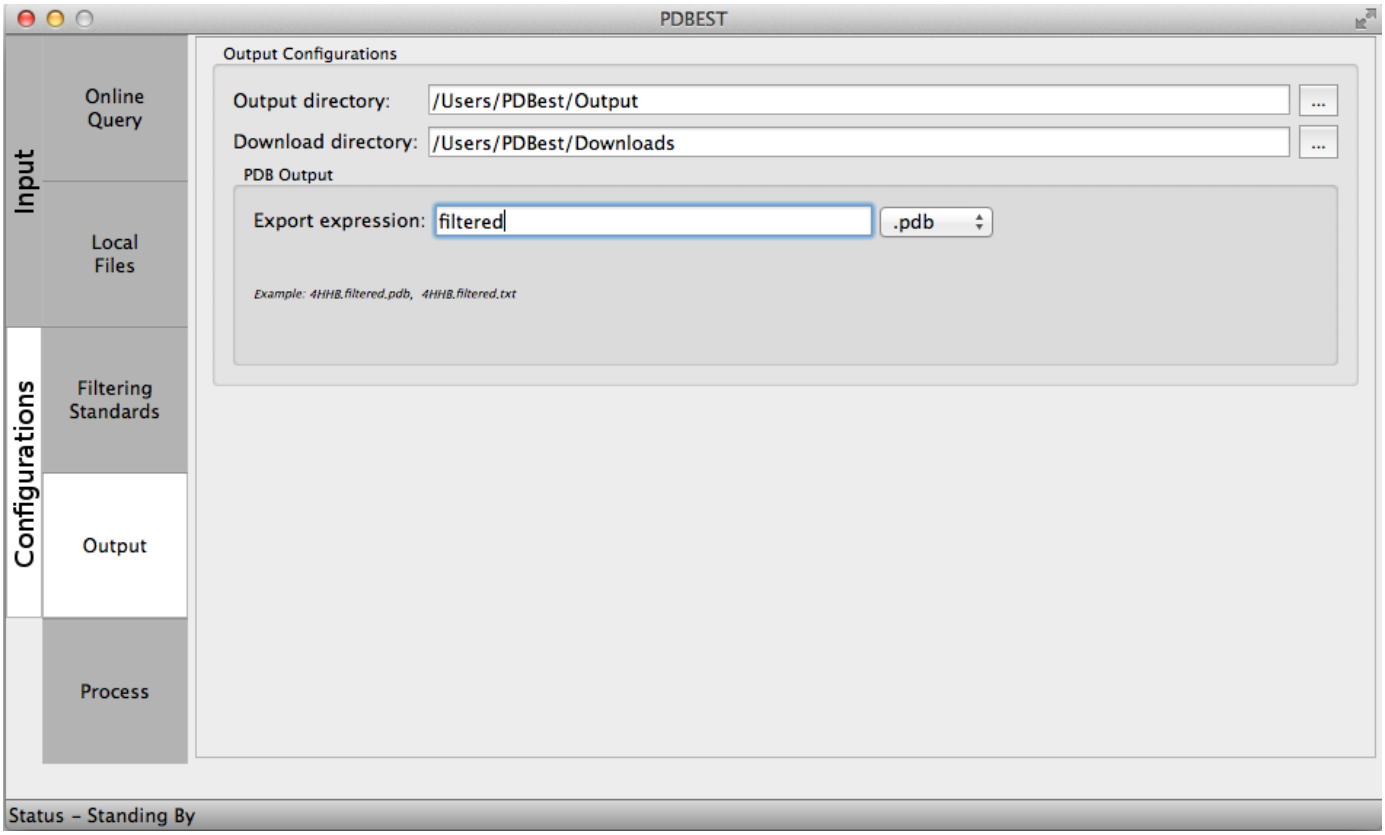


- **Bookkeeping :** The bookkeeping section provides final information about the file itself. It is necessary to check the MASTER and END boxes to keep these records.



Output

The Output configurations allow users to select the directory where the PDB files will be download from PDB online repository if online queries are performed and where processed files will be stored after the application of filters. The user may include an expression and choose the output format. For example: using the expression *".filtered"* every processed file will contain this term as in 4HHB.filtered.pdb.



Process

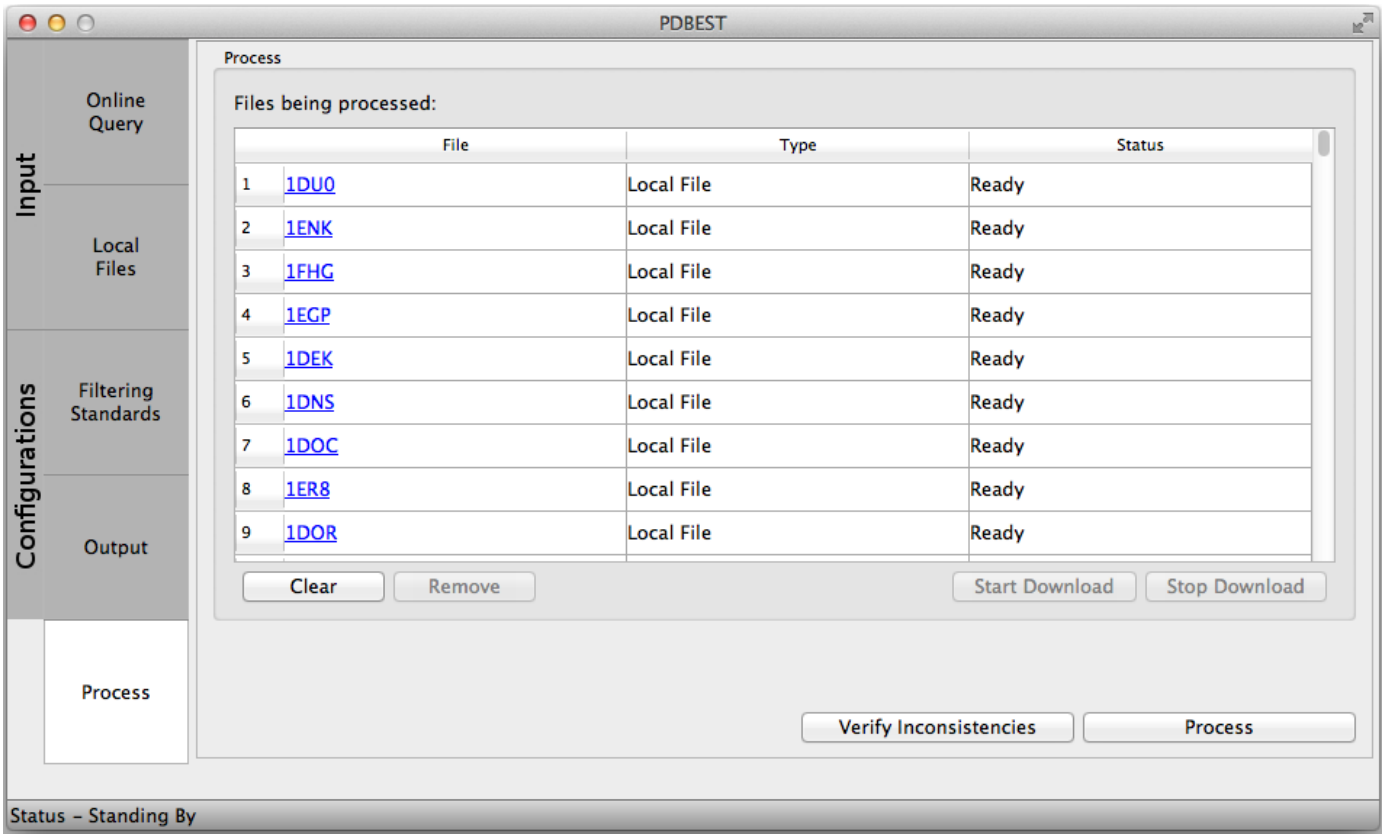
After submiting queries to RCSB Protein Data Bank (http://pdb.org) and/or loading local files and choosing filtering criteria, the files can be seen on a grid on the **Process** section. Up to a hundred file indentifiers can be seen at once. On the grid the origin of the file (online/local) and its status (to be downloaded/ready) is shown. The files with "To be downloaded" status will be downloaded by pressing the **"Start Download"** button.

Files can be selected and removed from the grid before processing.

Users can verify inconsistencies on files format with the **Verify Inconsistencies** button. Files with errors are marked on the grid. The possible issues on the structures verified by PDBest are missing atoms or residues, atoms with multiple occupancy and non-standard residues .

The **Open Detailed file** button will open a window with a complete report regarding the inconsistencies found. The report can lso be saved into a file.

The **"Process"** button applies the selected filters, generating the new files.



Av. Antônio Carlos, 6627, CEP 31270-010, Belo Horizonte, MG, Brazil, Telefone: +55 31 3499 5896

Estudo de Caso

BIOINFORMATICS APPLICATIONS NOTE

Vol. 00 no. 00 2015
Pages 1–3

Supplementary Material for *"PDBest: a user-friendly platform for manipulating and enhancing protein structures"*

Wellisson R. S. Gonçalves[1*], Valdete M. Gonçalves-Almeida[1], Aleksander L. Arruda[1], Wagner Meira Jr.[1], Carlos H. da Silveira[2], Douglas E. V. Pires[3*†], Raquel C. de Melo-Minardi[1*†]

[1]Department of Computer Science, Universidade Federal de Minas Gerais, Brazil
[2]Advanced Campus at Itabira, Universidade Federal de Itajubá, Brazil
[3]Centro de Pesquisas René Rachou, Fundação Oswaldo Cruz, Brazil

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

1 FIGURE

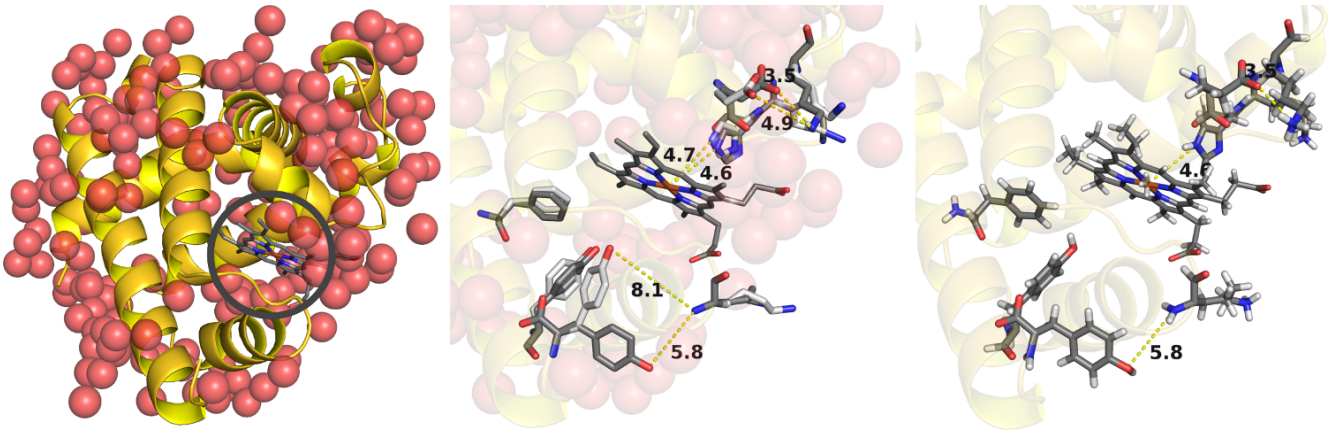


Fig. 1. PDBest - a freely available, user-friendly platform for manipulating protein structure files. The figure showcases an example of common filtering tasks that can be performed with PDBest: adding hydrogens, removing solvent molecules and multiple occupancies. The figure (generated using Pymol) depicts the structure of the sperm whale myoglobin (PDB:1A6M - on the left), the residues surrounding its bound Heme group (center), and the final processed file (on the right).

*To whom correspondence should be addressed
†These authors contributed equally to this work

2 PDBEST: GENERAL FILTERING PROTOCOL

The PDBest platform has emerged from our group's experience acquired over the year with PDB files and the need for special file filtering capabilities and reproducibility that go beyond the functionalities currently available online at the wwPDB and other tools.
We have elaborated a general PDB filtering protocol using PDBest, that encompass the most important and general pre-processing tasks that might help users to build a curated structural database. The main contemplated tasks on this tutorial are described below:

- Select structures based on experimental method (e.g, X-ray crystallography only);
- Identify non-protein chains;
- Remove identical chains (similar sequences at 100%);
- Select structures based on X-ray resolution and r-value;
- Split files by chain;
- Filter models;
- Identify chains containing missing atoms or residues;
- Filter atom with multiple occupancies (keep only the greater one);
- Convert between file formats (e.g, PDB, mmCIF, FASTA);
- Remove water molecules;
- Remove ANISOU record type;

The general protocol file is available at:
http://www.pdbest.dcc.ufmg.br/protocol/general-protocol.prt

3 CASE STUDIES

In this section we describe successfully developed studies by our group and co-workers where the use of PDBest was key.

3.1 Case #1: Studying contacts in globular proteins

Studies about how to reliably prospect inter-residue contacts in proteins using diferent methodologies have been performed by our group [1] and co-works. In this work the authors conducted a comparative analysis between two classical approachs: the traditional cutoff dependent (CD) and cutoff free Delaunay tessellation (DT). A database was built, comprising three main protein structural classes: all alpha, all beta, and alpha/beta. The following general selection criteria was performed with PDBest:

1. **Online Query:** Select on the SCOP Classification Browser the classes *all alpha* OR *all beta* OR *alpha/beta* AND X-ray resolution $\leq$ 2.0Å AND Refinement R Factor *R-Work* $\leq$ *0.2*. Also remove similar sequences with a 30% similarity threshold. Only select chains containing from 50 to 600 amino acid residues.
2. **Download and verify inconsistences:** After using the query presented in the previous step, the PDB files downloaded were examined. Files with potentially harmful inconsistencies which could introduce biases to the contact analysis were discarded. These included gaps on the structure and missing atoms.
3. **Pre-Process:**

    - Re-enumerate residues and atoms;
    - Exclude chains with the same sequence (as given by the SEQRES record);
    - Delete chains with missing atoms;
    - Discard chains with non-standard amino acids (with the exception of selenomethionine);
    - Only keep atoms with greatest occupancy on Coordenate Section;
    - Keep only the first model on Coordenate Section.

Protocol file available at:
http://www.pdbest.dcc.ufmg.br/protocol/globin-protocol.prt

3.2 Case #2: Studying Cross-Inhibition in Serine Proteases

In a previous work of cross-inhibition [2], the PDBest plataform was used successfully to query the Protein Data Bank and pre-process PDB files of complexes between Serine Proteases and the inhibitor Eglin C. The goal of the work was to study structural patterns on the contact interface between the protein and its ligands and understand why different proteins are inhibited by the same proteic ligand, Eglin C. Several steps were used to acquire and treat a set of target molecules in the study. The steps are described below:

1. **Online Query:** Query structures with macromolecule name *"Eglin C"* and two assimetric chains, with X-ray resolution $\leq$ 2Å.

2. **Download and verify inconsistences:** The PDB files were downloaded and examined to discover possible annotation or submission errors. PDBest provides a report indicating missing atoms and residues apart from the identification of multiple occupancies and non-standard residues. It is possible to decide to keep or discard the files based on this report.
3. **Pre-Process:** The PDB files were filtered considering the following set of requirements:

    - Delete hydrogens;
    - Split files by chain;
    - Keep atoms and residues original numbering on the Coordenate Section;
    - If multiple occupancies are identified, keep only the greater one;
    - Remove solvent molecules, HETATM and anisotropy records on the Coordenate Section.

Protocol file available at:
http://www.pdbest.dcc.ufmg.br/protocol/cross-inhibition-protocol.prt

3.3 Case #3: Molecular Recognition on Protein Kinases (CDK)

Noncovalent interactions are driving forces guiding the molecular recognition between proteins and ligands. Particularly, for the CDK (Cyclin-Dependent Kinase) protein family this phenomenon have attracted great interest due to binding site promiscuity. In other words, a vast number of diferent ligands bind the same binding site. In a recent work developed and presented at the ISCB-Latin American [3] this phenomenon was analised through a visual approach and PDBest was essential to query and pre-process a set of complexes deposited in the PDB.

1. **Online Query:** Perform a text search using *"CDK2 in complex with inhibitor RC"*.
2. **Download and verify inconsistences:** The PDB files were downloaded and examined to discovery some possible annotation or submission errors. The PDBest provides a report indicating about missing atoms and residues besides to identify occupancy and no-standard residues. In this case, all files presenting any missing residues were discarded and multiple occupancies repaired as described in the next step.
3. **Pre-Process:**

    - Delete hydrogens;
    - Split files by chain;
    - Keep atoms and residues original numbering on the Coordenate Section;
    - If multiple occupancies are identified, keep only the greater one;
    - Remove solvent molecules and anisotropy records on the Coordenate Section.

Protocol file available at:
http://www.pdbest.dcc.ufmg.br/protocol/protein-kinase-protocol.prt

REFERENCES

[1]C H da Silveira, D E V Pires, R C Melo-Minardi, C Ribeiro, C J M Veloso, J C D Lopes, W Meira Jr, G Neshich, C H I Ramos, R Habesch, and M M Santoro. Protein cutoff scanning: A comparative analysis of cutoff dependent and cutoff free methods for prospecting contacts in proteins. *Proteins: Struct., Funct., Bioinf.*, 74(3):727–743, 2009.

[2]V M Gonçalves-Almeida, Douglas E V Pires, Raquel Cardoso Melo-Minardi, Carlos Henrique da Silveira, W Meira Jr., and Marcelo M Santoro. HydroPaCe: understanding and predicting cross-inhibition in serine proteases through hydrophobic patch centroids. *Bioinformatics*, 28(3):342–349, 2012.

[3]A V Fassio, S A Silveira, and R C Melo-Minardi. Visual and interactive strategies to reveal patterns of protein-ligand interactions. In *ISCB-Latin American*, 2014.