

ドッキングシミュレーションにおける 試行錯誤的なパラメータ調整支援システム

前野 隆志[†] 伊達 進[†]
木戸 善之^{†,‡} 下條 真司^{†,‡}

今日、計算機を利用した薬物候補化合物のスクリーニングに関する期待と関心が高まっている。スクリーニングで利用されるドッキングシミュレーションでは、パラメータ数の多さや関連するツール間の依存によって、パラメータ調整が複雑になっている現状がある。そのため研究者は、複数のツールを用いたパラメータ調整を試行錯誤的に行わなければならない現状がある。本稿では、ツール・パラメータの選定・設定を柔軟な階層構造で表現できる「試行セット」と呼ばれるモデルを表現したデータ構造を用いて、ツール実行を効率よく実施するための支援システムを提案する。本システムによりドッキングシミュレーションを利用したスクリーニング全体の効率性の向上が見込まれる。

A System Assisting Trial-and-Error Parameter Tuning for Docking Simulation

TAKASHI MAENO,[†] SUSUMU DATE,[†] YOSHIYUKI KIDO,^{†,‡}
and SHINJI SHIMOJO^{†,‡}

Molecular docking simulation is a promising application for drug design. Although docking simulation and the related information technology have advanced in recent years, scientists still have difficulty in finding a suitable parameter set of docking programs because parameter-tuning step contains the scientists' trial-and-error processes. We define "trial set" model that can flexibly describe tools' flow and a way of parameter tuning for that tools. In this paper, we propose a system that can perform procedures described in a trial set to assist scientists' trial-and-error in parameter tuning. This system has potential to improve the whole process of *in-silico* screening.

1. はじめに

今日、計算機を利用した薬物候補化合物のスクリーニングに関する関心と期待が高まっている。ドッキングシミュレーションによりタンパク質と化合物の結合過程を計算機上で表現することで、薬物候補となり得るものとそうでないものを判定することができる。創薬研究者は、ドッキングシミュレーションにより算出されたスコア値を基準にタンパク質と化合物の結合強度を比較することができる。

ドッキングシミュレーションを実現するアプリケーションプログラムは、DOCK^{1),2)}, AutoDock³⁾,

GOLD⁴⁾ など多数存在する。今日これらのドッキングシミュレーションを応用し、さまざまなタンパク質および化合物に対する適合度に関する比較・評価が報告されており^{5),6)}、創薬に携わる研究者らの当該シミュレーションに対する期待はますます高まる傾向にある。DOCKは古くから知られているドッキングシミュレーションであり、営利目的でない学術的な研究においては無償利用でき、ドッキングシミュレーションを実施する際に必要となる一連のツール群が DOCK suite として提供されており、多くの研究者が注目しているツールの一つである。

ドッキングシミュレーションをはじめとしたシミュレーションを効率的に実行するための技術として、複数の計算機を利用して高速に計算を行う High Performance Computing (HPC) 技術が注目されている。クラスタコンピューティング技術は代表的な HPC 技術であり、Portable Batch System (PBS)⁷⁾ や Sun Grid Engine (SGE)⁸⁾ などのローカルスケジューラ

[†] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology,
Osaka University

[‡] 三井情報開発株式会社
Mitsui Knowledge Industry Co., Ltd

^{††} 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University

の利用により大量のジョブをクラスタ上の複数の計算機に割り振ることができる。さらに、近年注目されてきているグリッド技術⁹⁾を用いることで、一組織の管理領域を超えた Virtual Organization (VO)¹⁰⁾ という枠組みの中で、さまざまな場所に分散して存在する計算機を効率よく利用することができる。

このようにクラスタ、グリッドなどの HPC 技術の進展の中で、それらの技術を利用する実行環境も複雑化し、システムを利用する研究者の負担が増大する傾向がある。このため、例えば創薬研究者があるシミュレーション結果から、新しいパラメータを設定したシミュレーションを実行するといった、判断を必要とするシミュレーション全体の効率化を図るためには、研究者自身の判断を効率よく実行環境で実現できる必要がある。本稿では、DOCK suite において創薬研究者が試行錯誤的にパラメータ調整方針を判断していかなければならない現状の問題を取り上げ、そのようなパラメータ調整を支援するシステムを述べる。

2. DOCK suite における試行錯誤的なパラメータ調整

本節では、DOCK suite の概要を述べた後、創薬研究者が行う試行錯誤的なパラメータ調整の問題について述べる。

2.1 DOCK suite

DOCK suite に含まれるツール群とその処理の流れを図 1 に示す。DOCK suite では、ドッキングシミュレーションを行う DOCK の他、DOCK の前処理を行うツールが含まれる。特に SPHGEN や GRID は、DOCK の前処理を行う重要なツールである。

SPHGEN はターゲットタンパク質の活性部位から、その化学的に反対の性質を表現した球を生成し、その一連の球の集合を表現した site file を出力する。化学的に反対の性質を持ったものは強く引き合うことが知られており、化合物の各部位は同じ性質を持つ球の近くに配置される可能性が高い。この前提から DOCK は、この site file を利用して化合物に含まれる各原子の配置を決める。

GRID は、タンパク質側の立体構造データ (receptor file) から相互作用エネルギー値を算出するためのファイルである grid file を生成する。grid file は、タンパク質の活性部位が持つ化学的性質を 3次元の格子状に集約した各点の情報を保持するファイルである。DOCK でのドッキングシミュレーションでは、site file 利用して化合物を配置するが、タンパク質との相互作用エネルギーはこの grid file に含まれる格子点の情報

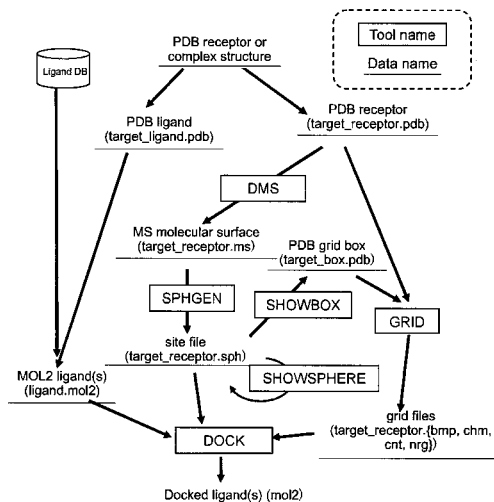


図 1 DOCK suite の流れ
Fig. 1 Flow of DOCK suite

を利用して算出する。

DOCK では、SPHGEN、GRID で出力された site file、grid file を利用してドッキングの対象となる化合物 (ligand file) との相互作用エネルギー値を計算し、そのエネルギー値や結合された化合物の構造等の結果を出力する。ターゲットタンパク質の立体構造から得られた site file と grid file が用意され、化合物のドッキングに関わるパラメータ (DOCK のパラメータ) が適切に設定されると、DOCK を用いた化合物のスクリーニングが可能となる。

SPHGEN、GRID、DOCK の 3つのツールに加え、SPHGEN のための前処理に位置する DMS、SPHGEN での結果の可視化のための SHOWSPHERE、GRID の前処理に位置する SHOWBOX などツールが存在する。これら全てのツールが連携して、DOCK によるドッキングシミュレーションが実現される。

2.2 試行錯誤的なパラメータ調整の問題

ドッキングシミュレーションを正しく行うためには、図 1 に含まれる全てのツールに対して適切なパラメータを設定する必要がある。DOCK suite におけるパラメータ調整過程では、試行錯誤的に各ツールのパラメータの調整を行わなければならない現状が存在する。

DOCK suite におけるパラメータ調整過程が試行錯誤的となる第 1 の要因としては調整すべきパラメータ数の多さと各パラメータ間の依存が挙げられる。DOCK suite の中でも GRID および DOCK のパラメータは特に多く、GRID で 21 個のパラメータ、DOCK パー

ジョン4で77個のパラメータ、DOCKバージョン5では96個のパラメータが存在する。

複数のパラメータが相互に実行結果に影響する場合には、パラメータ1つだけを変動させて、パラメータ調整を行うことは難しい。その場合、関係する他のパラメータ群との影響を考慮しながらパラメータ調整を行う必要がある。DOCK suiteにおけるパラメータ調整では、多数のパラメータの中から相互に依存すると判断したパラメータを適宜選び出してパラメータ調整を行い、その調整結果を見ながら次のパラメータ調整方針を決定するといった試行錯誤的な繰り返しが行われる。

第2の要因としてツール間の依存性が挙げられる。DOCK suiteは、図1で示したようにツール間で相互に連携の関係がある。そのため、例えばGRIDのパラメータ調整の正当性を確認するためにはDOCKによる最終的なドッキングシミュレーションの結果が必要である。このように、いくつかのツールが関連して動作するツール群に対するパラメータ調整では、いくつかのツールをまとめて実行して調整が行わなければならない。

DOCK suiteにおけるパラメータ調整例を表1に示す。表1はDOCKの96のパラメータの内(DOCKバージョン5の場合)、2つ(Orientation, Flexible)をピックアップしてパラメータ調整を行う例である。Orientationはドッキングシミュレーションでの化合物の並進・回転数を決めるパラメータであり、Flexibleは化合物の変形の回数に関わるパラメータとなる。表中のScoreとRMSDが結果を表している。

表1のパラメータ調整では、創薬研究者は並進・回転や変形の回数(2つのパラメータ値)ができる限り少なく、かつScoreとRMSDがともに小さくなるパラメータの組み合わせを探索する。このようなパラメータ調整方針は、研究者の経験と知識に基づいて決定される。表1のパラメータ調整では例えば、Orientation, Flexibleをそれぞれ100, 100が決定される。表1で示すようなパラメータ調整を繰り返し、全てのパラメータの最適値を確定していく必要がある。表1のパラメータ調整方針を、本研究では図2のようなモデルで表現する。本稿以降では、パラメータ調整に必要とされる解析空間を示す図2のようなモデルを「試行セット」として定義する。

表1でのパラメータ調整では、下2行の結果では回数が増えると結果が悪化している(Score, RMSDがともに増加している)。このような場合、その他のパラメータやその前処理のツールでのパラメータが適切

表1 パラメータ調整例
Table 1 An Example of Parameter Tuning

Orientation	Flexible	Score	RMSD (Å)
1	25	-22.25	10.47
1	100	-21.06	10.33
10	25	-30.83	1.87
10	100	-31.48	1.59
100	25	-28.84	1.68
100	100	-38.19	1.15
1000	25	-31.59	1.80
1000	100	-32.46	1.56

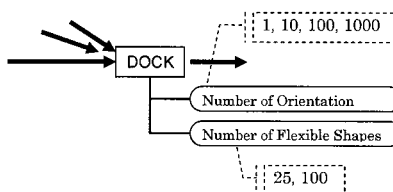


図2 表1のパラメータ調整のモデル(試行セット)
Fig.2 Parameter Tuning Model Corresponding to Table 1

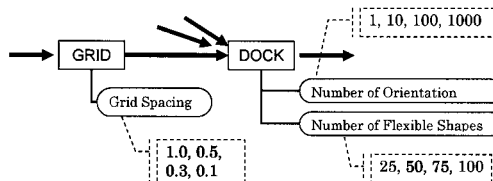


図3 試行錯誤の過程で変更されたパラメータ調整の例
Fig.3 An Example of Parameter Tuning Modified through Trial-and-Error Process

に設定されていない可能性がある。創薬研究者は図2(表1)でのパラメータ調整結果から、例えばDOCKの前のGRIDにおけるパラメータの1つである“Grid Spacing”のパラメータ調整が必要であると判断した場合、図3に示すようにGRIDも含めた2つのツールを利用したパラメータ調整を行う必要がある。

以上のように、DOCK suiteにおけるパラメータ調整では、利用するツールやパラメータ変動方法を適宜変化させて結果を確認し、次のパラメータ調整方針を決定するという、試行錯誤的な作業が繰り返し行われる。本研究ではこのような試行錯誤の繰り返しの中で変化する、研究者の考えるパラメータ調整実施方針をシームレスに実行環境で実現できる支援環境を提案・構築する。

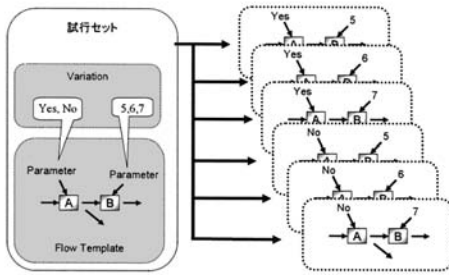


図 4 試行セットの例とその中に含まれるフロー群

Fig. 4 An Example of Trial Set and Flows in the Trial Set

3. 試行錯誤的なパラメータ調整支援システム

3.1 パラメータ調整に必要な解析空間を表現するデータ構造

図 4 は図中「A→B」の 2 つのツールで構成されるフローに対して、ツール A のパラメータを“Yes”，“No”の 2 種類，ツール B のパラメータを“5”，“6”，“7”の 3 種類に変化させるといった変動情報を加えた試行セットを表している。すなわち図 4 で表される試行セットの場合，各ツールに与えるパラメータ値が異なる 6 種類の「A→B」のフローが含まれる。

著者らは，図 4 に示すようなモデルを表現するために図 5 の Entity Relationship Diagram (ERD) で示される階層構造を持たせたデータ構造を定義している¹¹⁾。このうち Flow Template に当たる階層でツールのフロー情報を保持し，Flow Group の階層のデータ構造でパラメータの変動情報を格納する。例えば ACTIVITY, INPUT, OUTPUT はそれぞれツール，入力データ，出力データの情報を格納するためのテーブルであり，TRANSITION は各ツールのつながりを表現するためのテーブルである。このツールのつながりを表現する TRANSITION に対してパラメータ変動の情報が関連付けられること(図中 VARIATION テーブル)，試行セットに含まれるフローの集合を表現することができる。

さらに図 5 に示すテーブルにはそれぞれ，DELETE_FLAG と呼ばれる属性を持たせている。DELETE_FLAG は不要になったツールやパラメータ変動情報などを区別するためのフラグとなる。DELETE_FLAG を利用することで，例えば解析の必要がない範囲のツール実行と実行すべきツールの判別ができる。

3.2 システム構築

試行錯誤的なパラメータ調整を支援するために，本研究では創薬研究者のパラメータ調整実施方針を表現

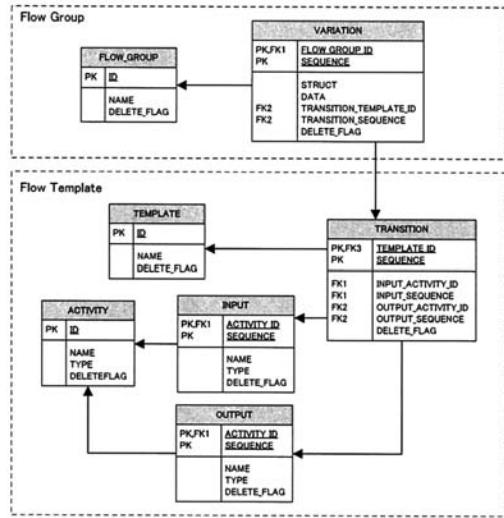


図 5 試行セットを表現するデータ構造

Fig. 5 Data Structure for Trial Set

した試行セットをシームレスに実行環境で実現できるシステムを提案する。システムコンセプトとして以下の 2 つの特徴を持つシステムを考える。

- 試行セットの解析：
図 5 ので示される試行セットのデータ構造に基づきパラメータ管理とフロー管理の分離を行い，試行セットに含まれるパラメータ調整実施方針に必要なツール実行やフローを解析する。
- 試行セットに含まれるツール実行：
試行セットに含まれる実施方針を既存の High-Performance Computing (HPC) 環境で効率よく実行する

本研究では問題解決のための利用者支援環境として図 6 に示すシステムアーキテクチャを提案する。システムは，“Presentation Layer”，“Problem Solving Environment (PSE) Layer”，“Grid Layer”の 3 つのレイヤで構成される。Presentation Layer はシステム利用者である創薬研究者のユーザインタフェース部分にあたる。PSE Layer は試行セットに含まれるツール群，パラメータ変動情報から必要となるツール実行を処理するレイヤとなる。また，Grid Layer は，具体的なツール実行を処理するレイヤである。

PSE Layer と Grid Layer の分離により，それぞれのレイヤの役割を明確に分離することができる。例えば PSE Layer では試行セットに含まれる手順の解析と管理を行うだけでよく，実際にどの計算機を利用して実行を行うといった具体的な実行処理の機能は必要

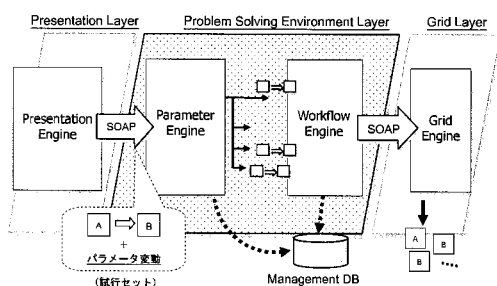


図 6 試行錯誤的な手順を支援するシステムアーキテクチャ
Fig. 6 System Architecture Assisting Trial-and-Error Processes

としない。一方、Grid Layer ではツールの流れやパラメータ変動情報などの手順に関する部分の管理を行う必要はなく、個々のツール実行を高速に実行することが要件として満たされればよい。そのため、グリッドやクラスタコンピューティングを始めとした既存の HPC 技術を有効に利用することができる。

PSE Layer にはパラメータ変動情報を解析する Parameter Engine とツールのフローを管理する Workflow Engine の 2 種類のエンジンが存在する。2 つのエンジンは図 5 で定義されたデータ構造を持つデータベースにアクセスし、データベース上に格納されている試行セットの解析を行う。著者らはこの PSE Layer を中心に実装を行った。

各エンジンの実装には、Java Development Kit (JDK)¹²⁾ バージョン 1.4.2 を利用し、アプリケーションサーバには Tomcat¹³⁾ バージョン 4.1.24, Web サービスエンジンとしては Axis¹⁴⁾ バージョン 1.1 を採用した。図 6 の PSE Layer で利用される Management DB には、図 5 のデータ構造に従って格納される。なお本実装では、リレーショナルデータベースシステムの一つである PostgreSQL¹⁵⁾ バージョン 7.2.1 を用いているが、他のバージョンや MySQL¹⁶⁾ などの他のデータベースを利用することもできる。

試行セットが図 6 に示すシステムに投入されると Parameter Engine が試行セットに含まれるパラメータ変動情報を解析し、実行が必要なフローの集合を生成する。個々のフローは Workflow Engine に送出され、それらのフローを受け取った Workflow Engine は、フロー内に含まれるツールとその流れを解析し、実行が必要なツールを決定する。実際の実行は Grid Layer に含まれる Grid Engine で処理される。以上のアーキテクチャにより、研究者はパラメータ調整で行いたい一連の手順を試行セットという単位で管理・指

示することができ、ツール間のデータの受け渡しや複数のフローの生成を逐一指示する必要がなくなる。これにより、創薬研究者は本来行いたいパラメータ調整実施方針を決定することに集中することができる。

3.3 DOCK suite への適用例と有用性に関する考察

DOCK suite を使った実際のパラメータ調整の本システムへの適用例を示す。評価には SHP-1¹⁷⁾ と呼ばれるタンパク質データ (PDB ID: 1FBR) を利用した。実験には、表 2 に示すスペックで構成される計算機を利用し、DOCK のバージョンは 5.4.0 を利用した。

表 2 SHP-1 のパラメータ調整に用いた計算機のスペック
Table 2 Computers Used in Parameter Tuning for SHP-1

CPU	Intel Pentium III 1.40GHz (Dual)
Memory	1 GB
OS	Linux version 2.6

パラメータ調整の例としては、DOCK における 4 つのパラメータに対するパラメータ調整を考える。表 3 に、パラメータ調整における各パラメータの値と結果を示す。表内のパラメータ a, b, c, d は表の下に示す実際のパラメータである。RMSD (Å) と実行時間 (秒) が実際に DOCK から得られた結果となる。

本調整例でのパラメータ調整の方針としては、実際の化合物の位置からの誤差を表す RMSD の値をできる限り小さくし、かつ計算時間が比較的小さいものを研究者が判断してパラメータ調整を行うものとする。計算時間が小さいものを選ぶ理由としては、後のパラメータ調整や、スクリーニングにかかる時間を極力抑える必要があることが挙げられる。

表 3 の灰色の塗りつぶし部分はデフォルト値を表している。パラメータ調整は表 3 の上から順に行われ、デフォルト値である 1 行目の値から、パラメータ調整を行っていく。その中で、本例ではパラメータを一つずつ変動させながら RMSD と計算時間の組み合わせのうち、最適なものを探していくという流れとなる。

表 3 の t1, t2, t3, t4 は、あるパラメータの値が決定されたときの時刻を表す。また、時刻 t0, t1, t2, t3 ではそれぞれパラメータ調整情報を変更した新しい試行セットをシステムに投入されている点である。例えば、時刻 t0 ではパラメータ b, c, d の値はデフォルトのまま固定でパラメータ a のみを 20, 50, 100, 200 に変動させるといった内容の試行セットが投入される。時刻 t1 では、結果から計算結果にほとんど差がないことからデフォルト値の 100 の値をパラメータ a の値に

表 3 SHP-1 を用いた実際のパラメータ調整例
Table 3 Parameter Tuning Example using SHP-1 Data

Parameters				RMSD (Å)	Execution Time (sec)	Number of Workflows Remained
a	b	c	d			
100	100	250	250	-	-	-
20	100	250	250	5.11	4,245	3
50	100	250	250	4.41	4,971	2
100	100	250	250	4.11	4,202	1
200	100	250	250	4.11	4,178	0
100	200	250	250	4.82	14,637	2
100	50	250	250	4.18	952	1
100	20	250	250	4.18	298	0
100	20	100	250	4.18	271	2
100	20	25	250	3.22	236	1
100	20	10	250	3.69	204	0
100	20	10	100	3.72	179	2
100	20	10	25	3.96	91	1
100	20	10	10	5.81	90	0

Parameter a: num_anchor_oriens_for_growth
Parameter b: number_confs_for_next_growth
Parameter c: simplex_anchor_max_iterations
Parameter d: simplex_grow_max_iterations
(Other parameters are set to the default values)

決定する。さらに時刻 t1 においては、パラメータ a, c, d を固定してパラメータ b を変動させる新しい試行セットを投入している。時刻 t2, t3 においても同様な作業を繰り返し、結果として本パラメータ調整でのパラメータ a, b, c, d の最適値として 100, 20, 10, 25 が決定された。

表 3 で示すパラメータ調整において、本章で提案するシステムを利用しない場合、研究者はそれぞれのパラメータの組み合わせ全てについて、DOCK を実行し、結果の取得・確認を行わなければならない。さらにそれぞれの実行では、具体的にツール実行に使用する計算機やツール間でのファイルの転送方法などについても、個々のツールの実行ごとに適宜指示して必要がある。一方、本システムの利用により、研究者はパラメータの変動情報を加えた試行セットを投入するだけで、必要なツールが自動的に実行される。また本パラメータ調整例では、一つのパラメータを数個変動させるだけの単純なものであるが、フローに含まれるツール数や同時に変動させるパラメータの数が増えるにつれて、ツールの流れの管理やパラメータの組み合わせが増大するため、本システムの役割は大きくなるものと考えられる。

以上のように、本研究で提案したシステムを利用することで、創薬研究者はどの計算機を使いどの順序でツールを逐次実行するか等の具体的な実行に関わる部分を考慮する必要がなく、パラメータの調整という本来研究者が行いたい作業に集中することができる。

4. 関連研究

本研究では、複数のツールの組み合わせ、パラメータの変動情報を含んだ研究者の考えるパラメータ調整実施方針をシームレスに実行環境で実現できる支援システムを構築している。本節では、本研究での内容と関連する他の研究について述べ、本研究の位置づけを明確にする。

4.1 ワークフロー

ツールの流れを管理するワークフローに関しては、多くの研究がなされている。ワークフローの標準団体として Workflow Management Coalition (WfMC)¹⁸⁾ が存在し、様々な標準化を行っている。WfMC では 5 つのインタフェースの定義を行っているが、そのうちワークフローの記述言語として XML Process Definition Language (XPDL)¹⁹⁾ の定義がされている。Kepler^{20),21)} はグリッド環境でのワークフローとして有名である。利用者の側で GUI を利用してワークフローを組み立てることが可能であることも Kepler の大きな特徴である。また、Gridbus Project²²⁾ におけるワークフローでは xWFL (XML-based workflow language) とよばれる、シンプルな仕様のワークフロー記述言語によりグリッド環境上でのワークフローを記述することができる²³⁾。ワークフローの研究は、主に複雑な処理の流れをまとめて記述することを目的として進められている。一方本研究では、試行セットとしてフローの中のツールを自由に組み立てられるだけでなく、パラメータ変動によるフローの集合を一まとまりで表現することができる。

4.2 パラメータ・スイープ

本研究では、ドッキングシミュレーション実施時における全体のスループット低下の要因が、研究者の試行錯誤的なパラメータ調整にあることを指摘している。ツールに対する最適なパラメータ値を探索するパラメータ・スイープの問題を扱っている研究も多く存在する。例えば、Nimrod²⁴⁾ や NASA の IPG²⁵⁾ では、Web インタフェースを通じてパラメータ調整方法を指定できる環境を提供している。その環境によりシステムの利用者は、変動させたいパラメータとその変動情報を Web インタフェースを通じて入力することで、システム上でパラメータの組み合わせが異なるツールが自動的に実行される。これらの研究では、パラメータ調整対象のツールが固定されているのが前提であり、主に単一のツールに対するパラメータ調整が対象となる。一方本研究は、パラメータ調整の際に利用するツール群を自由に組み合わせるパラメータ調整

が実現できる。また、パラメータ調整に係る一連の手順を、階層的なデータ構造により管理・実行ができることも本研究の大きな特徴となる。

5. まとめと今後の課題

本稿では試行錯誤的な繰り返しで変化する、複数のツールを利用したパラメータ実施方針をシームレスにシステム上で実行するためのシステムを提案し、その実装について述べた。さらに、DOCK suite におけるパラメータ調整過程の一例を取り上げ、提案する支援システムにより試行錯誤的な手順を実行する上での有効性を述べた。

本提案システムでは、パラメータの組み合わせによって複数のフローを生成し、これを下位の Grid Layer へシームレスに実行環境へ反映させることができるが、これをフロー生成時に研究者の与える優先度に応じて計算ジョブを管理する仕組みは今後の課題の1つとして挙げられる。このようなフローの優先度に応じた計算ジョブ管理機構の実現により、創薬に携わる研究者が関心のあるフローの実行結果を優先的に取得することが可能になり、より効率的な研究者支援環境が実現できると考えられる。

また、図6で示したシステムアーキテクチャのうち、Presentation Engine の具体的な実装は現時点では行われていない。より効率のよい利用者支援を目指すためには、例えば研究者が視覚的にツールを組み合わせるワークフローを作成し、その視覚的なワークフローに対してパラメータの変動情報を加えることで試行セットを作成するといった、ユーザインタフェース環境の充実が必要不可欠となると考えている。

参考文献

- 1) T. J. A. Ewing and I. D. Kuntz. Critical Evaluation of Search Algorithms for Automated Molecular Docking and Database Screening. *Journal of Computational Chemistry*, Vol. 18, No. 9, pp. 1175–1189, December 1998.
- 2) UCSF DOCK. <http://dock.compbio.ucsf.edu/>.
- 3) G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J. Olson. Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function. *Journal of Computational Chemistry*, Vol. 19, No. 14, pp. 1639–1662, November 1998.
- 4) G. Jones, P. Willett, R. C. Glen, A. R. Leach, and R. Taylor. Development and Validation of a Genetic Algorithm for Flexible Docking. *Journal of Molecular Biology*, Vol. 267, No. 3, pp. 727–748, April 1997.
- 5) M.D. Cummings, R.L. DesJarlais, A.C. Gibbs, V. Mohan, and E.P. Jaeger. Comparison of Automated Docking Programs as Virtual Screening Tools. *Journal of Medicinal Chemistry*, Vol. 48, No. 4, pp. 962–976, February 2005.
- 6) C. Bissantz, G. Folkers, and D. Rognan. Protein-Based Virtual Screening of Chemical Databases. 1. Evaluation of Different Docking/Scoring Combinations. *Journal of Medicinal Chemistry*, Vol. 43, No. 25, pp. 4759–4767, November 2000.
- 7) A. Bayucan, R. L. Henderson, J. P. Jones, C. Lesiak, B. Mann, B. Nitzberg, T. Proett, and J. Utley. *Portable Batch System Administrator Guide*. Veridian Systems, August 2000.
- 8) Grid Engine Project. <http://gridengine.sunsource.net/>.
- 9) I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, November 1998.
- 10) I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *The International Journal of High Performance Computing Applications*, Vol. 15, No. 3, pp. 200–222, Fall 2001.
- 11) 木戸 善之, 伊達 進, 前野 隆志, 長谷川 一郎, 下條 真司, 松田 秀雄. in-Silico スクリーニングを支援するワークフロースキーマの設計と実装. 情報処理学会第2回バイオ情報学研究会, pp. 17–24, October 2005.
- 12) Sun Microsystems, Inc. Java 2 Platform, Standard Edition (J2SE). <http://java.sun.com/j2se/>.
- 13) Apache Project. Apache tomcat. <http://tomcat.apache.org/>.
- 14) Apache Axis. <http://ws.apache.org/axis/>.
- 15) PostgreSQL. <http://www.postgresql.jp/>.
- 16) MySQL. <http://www.mysql.gr.jp/>.
- 17) J. Yang, Z. Cheng, T. Niu, X. Liang, Z. J. Zhao, and G. W. Zhou. Structural Basis for Substrate Specificity of Protein-tyrosine Phosphatase SHP-1. *Journal of Biological Chemistry*, Vol. 275, No. 6, pp. 4066–4071, February 2000.
- 18) The Workflow Management Coalition. <http://www.wfmc.org/>.
- 19) The Workflow Management Coalition. Workflow Process Definition Interface – XML Process Definition Language. Document Number WPMC-TC-1025, Version 2.00, October 2005.
- 20) I. Altintas, C. Berkley, E. Jaeger, M. Jones,

- B. Ludascher, and S. Mock. Kepler: An Extensible System for Design and Execution of Scientific Workflows. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM'04)*, pp. 21–23, June 2004.
- 21) Kepler Project. <http://kepler-project.org/>.
 - 22) The Gridbus Project. <http://www.gridbus.org/>.
 - 23) J. Yu and R. Buyya. A Novel Architecture for Realizing Grid Workflow using Tuple Spaces. In *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing (GRID 2004)*, pp. 119–128, November 2004.
 - 24) D. Abramson, R. Sasic, J. Giddy, and B. Hall. Nimrod: A Tool for Performing Parameterised Simulations Using Distributed Workstations. In *Proceedings of the 4th IEEE Symposium on High Performance Distributed Computing(HPDC-4)*, pp. 112–121, August 1995.
 - 25) M. Yarrow, K. M. McCann, R. Biswas, and R. vander Wijngaart. An Advanced User Interface Approach for Complex Parameter Study Process Specification on the Information Power Grid. In *Proceedings of GRID 2000: First IEEE/ACM International Workshop*, pp. 146–157, December 2000.