

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5868982号
(P5868982)

(45) 発行日 平成28年2月24日 (2016. 2. 24)

(24) 登録日 平成28年1月15日 (2016. 1. 15)

(51) Int. Cl.		F I	
GO 1 N 33/49	(2006. 01)	GO 1 N 33/49	A
GO 1 N 33/72	(2006. 01)	GO 1 N 33/49	Y
A 6 1 K 38/22	(2006. 01)	GO 1 N 33/72	A
A 6 1 K 33/26	(2006. 01)	A 6 1 K 37/24	
A 6 1 K 31/7135	(2006. 01)	A 6 1 K 33/26	

請求項の数 11 (全 46 頁) 最終頁に続く

(21) 出願番号 特願2013-529382 (P2013-529382)
 (86) (22) 出願日 平成23年9月16日 (2011. 9. 16)
 (65) 公表番号 特表2013-541712 (P2013-541712A)
 (43) 公表日 平成25年11月14日 (2013. 11. 14)
 (86) 国際出願番号 PCT/US2011/052038
 (87) 国際公開番号 W02012/037524
 (87) 国際公開日 平成24年3月22日 (2012. 3. 22)
 審査請求日 平成26年9月16日 (2014. 9. 16)
 (31) 優先権主張番号 61/383, 357
 (32) 優先日 平成22年9月16日 (2010. 9. 16)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 592017633
 ザ ジェネラル ホスピタル コーポレイ
 ション
 アメリカ合衆国 マサチューセッツ州 ボ
 ストン フルーツ ストリート 55
 (74) 代理人 100092783
 弁理士 小林 浩
 (74) 代理人 100120134
 弁理士 大森 規雄
 (74) 代理人 100104282
 弁理士 鈴木 康仁
 (72) 発明者 ヒギンズ, ジョン エム.
 アメリカ合衆国 マサチューセッツ州 O
 2 1 3 8, ケンブリッジ, サクラメント
 ストリート 42

最終頁に続く

(54) 【発明の名称】 診断のための赤血球動力学

(57) 【特許請求の範囲】

【請求項 1】

被験者が鉄欠乏性貧血 (I D A) を発症する危険を判定する方法であって、
 前記被験者の赤血球を含むサンプル中で、母平均赤血球ヘモグロビン濃度 (M C H C)
 を判定するステップと、

赤血球体積およびヘモグロビン含有量を M C H C 線上に投影することにより、各赤血球
 体積およびヘモグロビン含有量を指標に変換するステップと、

変換された体積およびヘモグロビン含有量指標が前記 M C H C 線上の平均投影位置の閾
 値百分率よりも低い前記サンプル中の赤血球分画を判定して、サンプル分画を提供するス
 テップと、

前記サンプル分画を基準分画と比較するステップとを含み、

前記基準分画よりも高いサンプル分画があれば、前記被験者が I D A を発症する危険が
 あることを示す方法。

【請求項 2】

消化器 (G I) 疾患をスクリーニングするために被験者をスクリーニングまたは選択す
 る方法であって、

前記被験者の赤血球を含むサンプル中で、母平均赤血球ヘモグロビン濃度 (M C H C)
 を判定するステップと、

赤血球体積およびヘモグロビン含有量を M C H C 線上に投影することにより、各赤血球
 体積およびヘモグロビン含有量を指標に変換するステップと、

変換された体積およびヘモグロビン含有量指標が前記MCHC線上の平均投影位置の閾値百分率よりも低い前記サンプル中の赤血球分画を判定するステップと、

前記サンプル分画を基準分画と比較するステップと、

前記サンプル分画が前記基準分画よりも低い場合、GI評価のために前記被験者を選択するか、またはさらにスクリーニングするステップとを含む方法。

【請求項3】

前記閾値百分率が、前記MCHC線上の前記母平均投影の70%、75%、80%、85%、90%、または95%である、請求項1または2に記載の方法。

【請求項4】

前記基準分画が、約0.10、0.11、0.12、0.13、0.14、または0.15である、請求項1または2に記載の方法。

10

【請求項5】

前記サンプルは、前記被験者の全血を含む、請求項1~4のいずれか一項に記載の方法。

【請求項6】

命令を記憶するためのメモリと、

動作を行う旨の前記記憶された命令を実行可能な1または複数のプロセッサまたは処理装置と

を有するコンピューティング装置を備え、前記動作が、

サンプル中の各赤血球の体積およびヘモグロビン濃度または含有量を表すデータを受信するステップと、平均赤血球ヘモグロビン濃度(MCHC)を表す線上に赤血球体積およびヘモグロビン含有量を投影することにより、各赤血球体積およびヘモグロビン含有量を指標に変換するステップと、

20

変換された体積およびヘモグロビン含有量指標が平均投影の閾値百分率よりも低い前記サンプル中の赤血球分画を計算するステップと、

前記分画を表す出力を提供するステップとを含む、システム。

【請求項7】

前記サンプル中の平均赤血球ヘモグロビン濃度(MCHC)を表すデータを受信する旨の命令をさらに含む、請求項6に記載のシステム。

【請求項8】

30

赤血球(RBC)の赤血球体積(CV)、平均赤血球体積(MCV)、血球ヘモグロビン濃度(CHC)、平均赤血球ヘモグロビン濃度(MCHC)、および平均赤血球ヘモグロビン含有量(MCH)の1または複数、例えば、すべてと、これらの母集団統計とを検出するように構成された検出モジュールをさらに備える、請求項6に記載のシステム。

【請求項9】

1または複数のプロセッサまたは処理装置により実行可能な動作を行う旨の命令を格納するように構成された、1または複数の機械読取可能な記憶装置であって、前記動作が、

各赤血球の体積およびヘモグロビン濃度を表すデータを受信するステップと、平均赤血球ヘモグロビン濃度(MCHC)を表す線上に赤血球体積およびヘモグロビン含有量を投影することにより、各赤血球体積およびヘモグロビン含有量を指標に変換するステップと

40

変換された体積およびヘモグロビン含有量指標が平均投影の閾値百分率よりも低い前記サンプル中の赤血球分画を計算するステップと、

前記分画を表す出力を提供するステップとを含む記憶装置。

【請求項10】

前記サンプル中の平均赤血球ヘモグロビン濃度(MCHC)を表すデータを受信する旨の命令をさらに含む、請求項9に記載の機械読取可能な記憶装置。

【請求項11】

(i)ヘモグロビン含有量減少率(D_h)の変動の大きさ、

(ii)赤血球体積減少率(D_v)の変動の大きさ、

50

- (i i i) 正規化臨界体積またはクリアランス閾値 (v_c)、
- (i v) 低速相体積およびヘモグロビン含有量の平均減少率 ()、
- (v) 高速相体積の平均減少率 (v)、ならびに
- (v i) 高速相ヘモグロビン含有量の平均減少率 (h)

のうちの1または複数を計算する旨の命令をさらに含む、請求項 6 ~ 8 のいずれか一項に記載のシステム、または請求項 9 または 10 に記載の機械読取可能な記憶装置。

【発明の詳細な説明】

【技術分野】

【0001】

優先権主張

本出願は、参照によりその全体が本明細書に組み込まれている、2010年9月16日に出願された米国仮出願第61/383,357号の優先権を主張する。

【0002】

連邦政府の支援による研究または開発

本発明は、国立衛生研究所により付与された認可番号第DK083242号および第HL091331号の下で、政府の支援により行われた。政府は本発明における確実な権利を有する。

【0003】

本発明は、貧血または貧血予備軍の患者を特定し、サラセミア形質貧血を鉄欠乏性貧血および他の原因の貧血と区別するための方法に関する。この方法を使用して、治療、例えば、鉄補給剤および/またはエリスロポエチン (EPO) 等の赤血球生成促進剤による療法を最適化することもできる。また、例えば前記方法に使用するための装置およびシステムについても記載する。

【背景技術】

【0004】

循環するヒト赤血球 (RBC) の数、大きさ、ヘモグロビン濃度、その他の特徴を制御するシステムについては、ほとんど解明されていない。RBCは骨髄から放出された後、未知の機構 (Lew VL他、(1995)、Blood 86、334-341、Waugh RE他、(1992)、Blood 79、1351-1358) により、体積および総ヘモグロビン含有量が減少し、約120日後に未知の要因に反応して除去される。

【発明の概要】

【発明が解決しようとする課題】

【0005】

統計物理学による理論および病院の臨床検査室からのデータ (d'Onofrio G他、(1995) Blood 85、818-823) の一部に基づき、本発明は、RBC成熟およびクリアランスのマスター方程式モデルを使用する。モデルは、貧血患者を正確に特定し、サラセミア形質貧血を鉄欠乏性貧血と区別する。驚いたことに、このモデルは、貧血が臨床的に検出可能になる数週間前に、多くの貧血予備軍患者を特定する。

【課題を解決するための手段】

【0006】

一態様では、被験者が鉄欠乏性貧血 (IDA) を発症する危険を判定する方法は、被験者の赤血球を含むサンプル中で、母平均赤血球ヘモグロビン濃度 (MCHC) を判定するステップと、赤血球体積およびヘモグロビン含有量をMCHC線上に投影することにより、各赤血球体積およびヘモグロビン含有量を指標に変換するステップとを含む。また方法は、変換された体積およびヘモグロビン含有量指標がMCHC線上の平均投影位置の閾値百分率よりも低いサンプル中の赤血球分画を判定して、サンプル分画を提供するステップと、サンプル分画を基準分画と比較するステップとを含む。基準分画よりも高いサンプル分画があれば、被験者がIDAを発症する危険があることを示す。

【0007】

別の態様では、消化器 (GI) 疾患をスクリーニングするために被験者をスクリーニン

10

20

30

40

50

グまたは選択する方法は、被験者の赤血球を含むサンプル中で、母平均赤血球ヘモグロビン濃度 (MCHC) を判定するステップと、赤血球体積およびヘモグロビン含有量を MCHC 線上に投影することにより、各赤血球体積およびヘモグロビン含有量を指標に変換するステップとを含む。また方法は、変換された体積およびヘモグロビン含有量指標が MCHC 線上の平均投影位置の閾値百分率よりも低いサンプル中の赤血球分画を判定するステップと、サンプル分画を基準分画と比較するステップとを含む。方法は、GI 評価のために被験者を選択するか、またはサンプル分画が基準分画よりも低い場合にさらにスクリーニングするステップをさらに含む。

【0008】

別の態様では、本開示は、小球性貧血を有する被験者、例えば、慢性疾患のない被験者の鉄欠乏性貧血 (IDA) とサラセミア形質 (TT) とを鑑別診断する方法を特徴とする。方法は、被験者の赤血球のサンプル中で、赤血球間のヘモグロビン含有量減少率 (D_h) の変動の大きさの値を判定するステップと、 D_h を基準値と比較するステップとを含む。基準値よりも高い D_h があれば、被験者が IDA であるか IDA になる可能性が高いことを示し、基準値よりも低い D_h があれば、被験者が TT であるか TT になる可能性が高いことを示す。

10

【0009】

別の態様では、健常者の血液ドーピングまたは赤血球生成促進剤の存在または使用を検出するための方法は、ヘモグロビン含有量減少率 (D_h) の変動の大きさ、および被験者のサンプル中の赤血球体積減少率 (D_v) の変動の大きさの一方または両方の値を判定するステップを含む。基準値よりも高い D_h および / または D_v があれば、血液ドーピングの存在もしくは使用または赤血球生成促進剤の使用を示す。

20

【0010】

別の態様では、被験者のエリスロポエチン (EPO) または他の赤血球生成促進剤 (ESA) の投与量を最適化するための方法は、EPO または ESA 治療を受けている被験者のサンプル中の正規化臨界体積 (v_c) を判定するステップを含む。また、方法は、 v_c が下位基準レベルよりも低い場合に EPO の投与量を増加させ、 v_c が上位基準レベルよりも高い場合に EPO の投与量を減少させるステップを含む。

【0011】

別の態様では、鉄剤を使用して被験者を治療する方法が、被験者のサンプル中の正規化臨界体積 (v_c) を判定するステップと、 v_c が下位基準レベルよりも低い場合に、1 回の鉄剤を投与するか、または鉄補給剤を処方するステップとを含む。

30

【0012】

別の態様では、本開示は、命令を記憶するためのメモリと、記憶された命令を実行可能な 1 もしくは複数のプロセッサまたは処理装置とを有するコンピューティング装置を備えたシステムを特徴とする。記憶された命令は、実行されると、サンプル中の各赤血球の体積およびヘモグロビン濃度または含有量を表すデータを受信するステップと、平均赤血球ヘモグロビン濃度 (MCHC) を表す線上に赤血球体積およびヘモグロビン含有量を投影することにより、各赤血球体積およびヘモグロビン含有量を指標に変換するステップとを含む動作を行う。また、動作は、変換された体積およびヘモグロビン含有量指標が平均投影の閾値百分率よりも低いサンプル中の赤血球分画を計算するステップと、分画を表す出力を提供するステップとを含む。

40

【0013】

別の態様では、本開示は、1 または複数のプロセッサまたは処理装置により実行可能な命令を記憶するように構成された、1 または複数の機械読取可能な記憶装置を特徴とする。命令は、1 または複数のプロセッサにより実行されると、サンプル中の各赤血球の体積およびヘモグロビン濃度を表すデータを受信するステップと、平均赤血球ヘモグロビン濃度 (MCHC) を表す線上に赤血球体積およびヘモグロビン含有量を投影することにより、各赤血球体積およびヘモグロビン含有量を指標に変換するステップとを含む動作を行う。また、動作は、変換された体積およびヘモグロビン含有量指標が平均投影の閾値百分率

50

よりも低いサンプル中の赤血球分画を計算するステップと、分画を表す出力を提供するステップとを含む。

【0014】

別の態様では、本開示は、(i)母平均赤血球ヘモグロビン濃度(MCHC)、赤血球体積およびヘモグロビン含有量をMCHC線上に投影することによる、各赤血球体積およびヘモグロビン含有量の指標への変換、投影が平均投影の閾値百分率よりも低いサンプル中の赤血球分画、(ii)ヘモグロビン含有量減少率(D_h)の変動の大きさ、(iii)赤血球体積減少率(D_v)の変動の大きさ、ならびに(iv)正規化臨界体積またはクリアランス閾値(v_c)の1または複数を計算する旨の機械読取可能な命令を含む、1または複数の機械読取可能な記憶装置を特徴とする。

10

【0015】

別の態様では、サラセミア形質(TT)について患者をスクリーニングする方法は、被験者のサンプル中のヘモグロビン含有量減少率(D_h)の変動の大きさの値を判定するステップと、範囲の参照のために D_h を比較するステップとを含む。 D_h が基準範囲外にあれば、患者がTTであるか、TTになる可能性があることを示し、かつ/あるいは、 D_h が基準範囲外にあれば、遺伝子型決定もしくはさらなる検査のために被験者を選択することを示し、または被験者が妊婦である場合には、その父親をスクリーニングのために選択することを示す。

【0016】

別の態様では、本開示は、貧血を有する(例えば、貧血と診断された)被験者において、慢性疾患(ACD)に伴う貧血と他の原因の貧血とを鑑別診断するための方法を特徴とする。方法は、(i)ヘモグロビン含有量減少率(D_h)の変動の大きさ、(ii)赤血球体積減少率(D_v)の変動の大きさ、(iii)正規化臨界体積またはクリアランス閾値(v_c)、(iv)低速相体積およびヘモグロビン含有量の平均減少率(\bar{v})、(v)高速相体積の平均減少率(\bar{v}_h)の1または複数の値を判定するステップと、これらの値を、ACDの診断確率が基準範囲内の値の数によって判定される基準範囲と比較するステップと、この確率を閾値と比較するステップと、確率が閾値よりも高い場合に、輸血療法をしない選択をし、かつ/または鉄レベルの追加検査を行わないことを選択するステップとを含む。

20

【0017】

別の態様では、本開示は、コンピュータ読取可能な命令を記憶するように構成されたコンピュータ読取可能な記憶装置を特徴とする。この命令は、1または複数のプロセッサにより実行されると、網状赤血球または赤血球の時間依存性体積-ヘモグロビン分布に関連する第1集合のパラメータを受信するステップと、第1集合のパラメータに基づいて赤血球の定常状態分布を推定するステップとを含む動作を行わせる。また、動作は、赤血球の推定定常状態分布と経験分布との相違の程度を表す目的関数を計算するステップを含む。

30

【0018】

別の態様では、本開示は、コンピューティング装置を備えたシステムを特徴とする。コンピューティング装置は、命令を記憶するためのメモリと、記憶された命令を実行して種々の動作を行うことのできる1または複数のプロセッサまたは処理装置を有する検出モジュールとを備える。動作は、網状赤血球または赤血球の時間依存性体積-ヘモグロビン分布に関連する第1集合のパラメータを受信するステップと、第1集合のパラメータに基づいて赤血球の定常状態分布を推定するステップとを含む。また、動作は、赤血球の推定定常状態分布と経験分布との相違の程度を表す目的関数を計算するステップを含む。

40

【0019】

別の態様では、コンピュータにより実施される方法が、網状赤血球または赤血球の時間依存性体積-ヘモグロビン分布に関連する第1集合のパラメータを受信するステップと、第1集合のパラメータに基づいて赤血球の定常状態分布を推定するステップとを含む。また、方法は、赤血球の推定定常状態分布と経験分布との相違の程度を表す目的関数を計算するステップを含む。

50

【 0 0 2 0 】

前述した方法、システム、コンピュータ読取可能な記憶装置の実施は、以下の特徴の組合せを含むことができる。

【 0 0 2 1 】

被験者の母平均体積

【 0 0 2 2 】

【数 1】

(\bar{v})

の 75% よりも大きい v_c 、または 78 ~ 82% の v_c を維持するように、EPO の投与量を調節することができる。被験者の母平均体積 10

【 0 0 2 3 】

【数 2】

(\bar{v})

の 78% よりも大きい v_c 、または 78 ~ 82% の v_c を維持するように、鉄剤の投与量を調節することができる。閾値百分率を、MCHC 線上の母平均投影の 70%、75%、80%、85%、90%、または 95% とすることができる。基準分画は、約 0.10、0.11、0.12、0.13、0.14、または 0.15 である。サンプルは、被験者の全血を含むことができる。サンプル中の平均赤血球ヘモグロビン濃度 (MCHC) を表すデータを受信することができる。検出モジュールを、赤血球 (RBC) の赤血球体積 (CV)、平均赤血球体積 (MCV)、血球ヘモグロビン濃度 (CHC)、平均赤血球ヘモグロビン濃度 (MCHC)、および平均赤血球ヘモグロビン含有量 (MCH) の 1 または複数、例えば、すべてと、これらの母集団統計とを検出するように構成することができる。体積およびヘモグロビン含有量座標が、すべての体積およびヘモグロビン含有量座標の最小二乗線形適合を表す線上に投影されるように、個々の赤血球の体積およびヘモグロビン含有量を変換することができる。 20

【 0 0 2 4 】

コンピュータまたは機械読取可能な記憶装置は、(i) ヘモグロビン含有量減少率 (D_h) の変動の大きさ、(ii) 赤血球体積減少率 (D_v) の変動の大きさ、(iii) 正規化臨界体積またはクリアランス閾値 (v_c)、(iv) 低速相体積およびヘモグロビン含有量の平均減少率 ()、(v) 高速相体積の平均減少率 (v)、ならびに (vi) 高速相ヘモグロビン含有量の平均減少率 (h) の 1 または複数 を計算する旨の命令を含むことができる。 30

【 0 0 2 5 】

目的関数の値が所定の閾値条件を満たす場合に、第 1 集合のパラメータを調節して第 2 集合のパラメータを提供することができる。第 2 集合のパラメータは、対応する推定定常状態分布と経験分布との相違の程度が低下するようになっている。第 2 集合のパラメータを、第 1 集合のパラメータとして 1 または複数のプロセッサに提供することができる。網状赤血球または赤血球の時間依存性体積 - ヘモグロビン分布は、患者から得た血液サンプルに基づくことができる。定常状態分布を、線形演算子と状態の測定した分布との組合せに基づいて推定することができる。線形演算子は、少なくとも 1 つのヤコビアンおよび少なくとも 1 つのラブラシアンを含むことができる。 40

【 0 0 2 6 】

特に定義のない限り、本明細書で使用されるすべての技術的および科学的用語は、本発明が属する技術の当業者が通常理解するものと同じの意味を有する。本発明で使用する方法および材料について本明細書で説明するが、当技術で公知の他の適切な方法および材料を使用してもよい。材料、方法、および例は例示的なものにすぎず、限定的なものではない。本明細書で言及するすべての刊行物、特許出願、特許、シーケンス、データベースエントリ、および他の参考文献は、参照により組み込まれている。矛盾がある場合には、定 50

義を含めて本明細書に従う。

【0027】

以下の詳細な説明、図面、および特許請求の範囲から、本発明の他の特徴および利点が明らかになる。

【図面の簡単な説明】

【0028】

【図1A】図1A及び1Bは、末梢循環における平均RBCの体積およびヘモグロビンの同時制御の経験的測定(1A)および動的モデル(1B)を示す図である。図1Aでは、網状赤血球分布が等確率密度等高線において実線で示され、全RBCの母集団が破線で示される。両パネルの起点へ突出する斜線は、母集団の平均細胞内ヘモグロビン濃度(MCHC)を表す。この線状のいずれかの位置にあるRBCは、MCHCと等しい赤血球ヘモグロビン濃度を有する。まず、高速動力学()により、各パネルの右上に示す典型的な大きい未熟網状赤血球についての体積およびヘモグロビンが減少する。次に、低速動力学()により、MCHC線に沿った体積およびヘモグロビンが減少する。生物学的プロセスには本質的にノイズがあるため、体積およびヘモグロビンの減少に必要な事象中の小さな不規則変動により、個々の赤血球ヘモグロビン濃度がMCHC線の周りでドリフトし、血球が除去されるときに臨界体積(図1Bの v_c)に達するまで、図1Bの挿入図に示すようにMCHC線の周りで大きさ(D)が変動し得る。

10

【図1B】図1A及び1Bは、末梢循環における平均RBCの体積およびヘモグロビンの同時制御の経験的測定(1A)および動的モデル(1B)を示す図である。図1Aでは、網状赤血球分布が等確率密度等高線において実線で示され、全RBCの母集団が破線で示される。両パネルの起点へ突出する斜線は、母集団の平均細胞内ヘモグロビン濃度(MCHC)を表す。この線状のいずれかの位置にあるRBCは、MCHCと等しい赤血球ヘモグロビン濃度を有する。まず、高速動力学()により、各パネルの右上に示す典型的な大きい未熟網状赤血球についての体積およびヘモグロビンが減少する。次に、低速動力学()により、MCHC線に沿った体積およびヘモグロビンが減少する。生物学的プロセスには本質的にノイズがあるため、体積およびヘモグロビンの減少に必要な事象中の小さな不規則変動により、個々の赤血球ヘモグロビン濃度がMCHC線の周りでドリフトし、血球が除去されるときに臨界体積(図1Bの v_c)に達するまで、図1Bの挿入図に示すようにMCHC線の周りで大きさ(D)が変動し得る。

20

30

【図2】20人の健常者と、3種類の軽度貧血を有する患者、すなわち、慢性疾患(ACD)に伴う貧血患者11人、サラセミア形質(TT)の患者33人、および鉄欠乏性貧血(IDA)の患者27人についてのモデルパラメータの一連の6つの箱型図である。各箱の上縁および下縁は、75パーセントイルおよび25パーセントイルに位置する。中央値は箱内部の水平線により示される。垂直線は、箱からの距離が四分位距離の1.5倍未満であるデータ点まで延びる。より極値のデータ点はプラス(+)記号として示される。高速動力学は、低速動力学は、不規則変動はD、およびクリアランス閾値は v_c により特徴付けられる。

【図3A】4か月後にIDAを発症した患者の全赤血球計算(CBC)の等高線図である。各図は、体積 - ヘモグロビン含有量確率密度の35%、60%、75%、および85%を囲む等高線を示す。起点からの破線はMCHCを表す。円は、各赤血球の体積 - ヘモグロビン含有量座標のMCHC線上への投影の平均値を示す。MCHC線に垂直な短い実線は、平均投影の85%に対応する線に沿った位置を標示する。図3Aは、患者がIDAを発症する116日前に測定された正常CBCを示す。計算された $P_{0.85}$ (グレーで塗りつぶされている)は正常である。図3Bは、IDA検出の65日後および51日前に測定された正常CBCを示す。CBCが正常であっても、 $P_{0.85}$ は異常である。パネルの図3Cは、IDAが診断されたときのCBCを示す。

40

【図3B】4か月後にIDAを発症した患者の全赤血球計算(CBC)の等高線図である。各図は、体積 - ヘモグロビン含有量確率密度の35%、60%、75%、および85%を囲む等高線を示す。起点からの破線はMCHCを表す。円は、各赤血球の体積 - ヘモグ

50

ロビン含有量座標のMCHC線上への投影の平均値を示す。MCHC線に垂直な短い実線は、平均投影の85%に対応する線に沿った位置を標示する。図3Aは、患者がIDAを発症する116日前に測定された正常CBCを示す。計算された $P_{0.85}$ （グレーで塗りつぶされている）は正常である。図3Bは、IDA検出の65日後および51日前に測定された正常CBCを示す。CBCが正常であっても、 $P_{0.85}$ は異常である。パネルの図3Cは、IDAが診断されたときのCBCを示す。

【図3C】4か月後にIDAを発症した患者の全赤血球計算（CBC）の等高線図である。各図は、体積 - ヘモグロビン含有量確率密度の35%、60%、75%、および85%を囲む等高線を示す。起点からの破線はMCHCを表す。円は、各赤血球の体積 - ヘモグロビン含有量座標のMCHC線上への投影の平均値を示す。MCHC線に垂直な短い実線は、平均投影の85%に対応する線に沿った位置を標示する。図3Aは、患者がIDAを

10

発症する116日前に測定された正常CBCを示す。計算された $P_{0.85}$ （グレーで塗りつぶされている）は正常である。図3Bは、IDA検出の65日後および51日前に測定された正常CBCを示す。CBCが正常であっても、 $P_{0.85}$ は異常である。パネルの図3Cは、IDAが診断されたときのCBCを示す。

【図3D】90日以内の第2の正常CBCを有する患者の20の正常CBCと、90日後までにIDAと診断された患者の20の正常CBCについての $P_{0.85}$ の箱型図である。 $P_{0.85}$ は、実際の診断の90日前まで、75%の感度および100%の特異性でIDAを順調に予測する。さらなる詳細について、以下および実施例3「鉄欠乏性貧血の予測」を参照されたい。

20

【図4】TTを有する5件の訓練ケース、IDAを有する5件の訓練ケース、TTを有する28件の検査ケース、IDAを有する22件の検査ケースについての D_h の分布を示す箱型図である。結果は、本方法が、小球性貧血の原因としてのTTとIDAとを区別できることを示す。

【図5】式3で説明するクリアランス確率を計算するために使用される投影距離（ \quad ）の概略図である。血球がMCHC線上に投影され、クリアランス確率は、この線に沿った投影点から閾値（ v_c ）までの距離の関数である。

【図6A】健常者の適合（濃いグレー）および経験的（薄いグレー）定常状態体積 - ヘモグロビン含有量確率分布の比較を示す3次元グラフである。図6Aは、MCHC線を通る垂直面に投影された図を示す（図1参照）。図6Bは、起点に向かって見た90度回転図

30

である。

【図6B】図6A及び6Bは、健常者の適合（濃いグレー）および経験的（薄いグレー）定常状態体積 - ヘモグロビン含有量確率分布の比較を示す3次元グラフである。図6Aは、MCHC線を通る垂直面に投影された図を示す（図1参照）。図6Bは、起点に向かって見た90度回転図である。

【図7】200を超える別個のシミュレーションによる最適化されたパラメータ値および残差平方和目的関数により判定された適合度の、一連の6つのヒストグラムである。目的関数のより小さい値は、より良い適合を意味する。図7は、すべてのモデルパラメータが、十分に定義された最適近傍を有することを示す。目的関数については式5を参照されたい。

40

【図8A】図2に示す「健常者」および「IDA」患者についての $P_{0.85}$ の分布を示す箱型図である。これらの結果に基づき、 $P_{0.85}$ の閾値0.121が選択されて、図3Dに示す患者の独立集合の検査に使用された。本図は、潜伏IDAの閾値を特定するための方法の例である。

【図8B】図8B～8Dは、図2に示す定常状態の健常者（各図の上パネル）およびIDA患者（各図の下パネル）についての $P_{0.85}$ （8B）、 $P_{0.75}$ （8C）、および $P_{0.90}$ （8D）の分布を示すヒストグラムである。

【図8C】図8B～8Dは、図2に示す定常状態の健常者（各図の上パネル）およびIDA患者（各図の下パネル）についての $P_{0.85}$ （8B）、 $P_{0.75}$ （8C）、および $P_{0.90}$ （8D）の分布を示すヒストグラムである。

50

【図 8 D】図 8 B ~ 8 D は、図 2 に示す定常状態の健常者（各図の上パネル）および I D A 患者（各図の下パネル）についての $P_{0.85}$ （8 B）、 $P_{0.75}$ （8 C）、および $P_{0.90}$ （8 D）の分布を示すヒストグラムである。

【図 9】図 9 A ~ 9 C は、表 1 A、B に挙げる単一 R B C 体積およびヘモグロビン含有量動力学の異なる関数形式の 3 つについての速度場を示す図である。黒の斜線は M C H C に等しい一定のヘモグロビン濃度を示す。

【図 10】図 10 A 及び 10 B は、表 1 A、B で説明する異なるクリアランス関数の 2 つについてのクリアランス確率を示す 3 次元グラフである。図 10 A は関数形式 A および C を示し、図 10 B は関数形式 B を示す。M C H C 線は黒で示される。

【図 11】20 人の健常者についての表 1 A、B からの関数形式 B に基づくモデルパラメータと、3 種類の軽度貧血を有する患者、すなわち、慢性疾患（A C D）に伴う貧血患者 11 人、サラセミア形質の患者 33 人、および鉄欠乏性貧血の患者 27 人についてのモデルパラメータの変化の様子とを示す、一連の 6 つの箱型図である。各箱の上縁および下縁は、75 パーセンタイルおよび 25 パーセンタイルに位置する。中央値は箱内部の水平線により示される。垂直線は、箱からの距離が四分位距離の 1.5 倍以内であるデータ点まで延びる。より極値のデータ点はプラス（+）記号として示される。高速動力学は、低速動力学は、不規則変動は D、およびクリアランス閾値は v_c により特徴付けられる。

10

【図 12】20 人の健常者についての表 2 からの関数形式 C に基づくモデルパラメータと、3 種類の軽度貧血を有する患者、すなわち、慢性疾患（A C D）に伴う貧血患者 11 人、サラセミア形質（T T）の患者 33 人、および鉄欠乏性貧血（I D A）の患者 27 人の変化の様子とを示す、一連の 6 つの箱型図である。各箱の上縁および下縁は、75 パーセンタイルおよび 25 パーセンタイルに位置する。中央値は箱内部の水平線により示される。垂直線は、箱からの距離が四分位距離の 1.5 倍以内であるデータ点まで延びる。より極値のデータ点はプラス（+）記号として示される。高速動力学は、低速動力学は、不規則変動は D、およびクリアランス閾値は v_c により特徴付けられる。

20

【図 13 A】図 13 A 及び 13 B は、本発明の一部の実施形態による、M C H C 線への投影時に体積およびヘモグロビン含有量がこの線に沿った平均投影位置の閾値百分率よりも低いサンプル中の、赤血球分画を判定するための例示的な方法ステップを示すブロック図である。

【図 13 B】図 13 A 及び 13 B は、本発明の一部の実施形態による、M C H C 線への投影時に体積およびヘモグロビン含有量がこの線に沿った平均投影位置の閾値百分率よりも低いサンプル中の、赤血球分画を判定するための例示的な方法ステップを示すブロック図である。

30

【図 14】本発明で使用する例示的なコンピューティング装置を示すブロック図である。

【図 15】患者の血液サンプルについての体積およびヘモグロビン含有量測定からのモデルパラメータを推測するプログラムのための例示的なコンピュータコードを示す図である。

【図 16】式 4 の数値近似を使用して定常状態 R B C 分布を計算する、呼び出された関数「 $s s R B C$ 」のための例示的なコンピュータコードを示す図である。

【図 17】被験者の血液サンプルについて $P_{0.85}$ を計算するプログラム「 $c a l c u l a t e C o m p e n s a t i o n$ 」のための例示的なコンピュータコードを示す図である。この関数は、他の関数をも呼び出すが、パラメータ計算コードから独立して実行され得る。

40

【発明を実施するための形態】

【0029】

赤血球は、ほとんど解明されていないプロセスにより循環から除去される。これらおよびその他の R B C 成熟およびクリアランスプロセスの根底をなす特定の分子および細胞機構は未知であるが、このようなプロセスは病気の状態において変化し、この変化によってこれらの病気を有する患者に種々の差を生じさせる可能性がある。例えば、体積またはヘモグロビン濃度のような特定の細胞特性が、循環における赤血球寿命中に変化する場合、

50

何らかの病気を有する患者の赤血球の母集団にわたるこの変化の平均速度が、病気がない患者の赤血球の母集団にわたるこの変化の平均速度と異なる可能性がある。

【0030】

健康な成人では、1日に約 2.5×10^{11} の新しいRBCが骨髄から末梢循環へ放出され、ほぼ同じ数が除去される。したがって、循環する母集団を有する血球は変化し続けるが、健常者（および軽度の病気を有する患者）では、母集団の特性が非常に安定している。臨床において、血液中の赤血球の体積分画（ヘマトクリット）、平均RBC体積（MCV）、RBC体積（RDW）の変動係数、および平均赤血球ヘモグロビン質量（MCH）等の母集団特性が、全赤血球計算（CBC）において定期的に測定される（1）。最近では、非常に若い（時間または日数が）循環RBC（網状赤血球）を特定し特徴付けることができるようになっている（2）。RBCは、骨髄から放出された後、数日で、体積およびヘモグロビンが急速に減少する（3）。この高速相に続いて、より緩やかに減少する、はるかに長い時間があり（4～7）、この間に体積およびヘモグロビンが同時制御される（8）。図1Aを参照されたい。

10

【0031】

網状赤血球および全循環RBC（図1A）の確率分布の比較は、体積とヘモグロビン含有量との相関が、血球の成熟とともに、網状赤血球母集団の初期相関係数約0.40から全母集団の約0.85まで増加することを示す。したがって、関与する分子機構の多くは未知であるが、図1Aおよび1Bの等濃度線に示すように、ヘモグロビン濃度が母平均赤血球ヘモグロビン濃度（MCHC）に向かう傾向があるように、平均RBCが成熟することが明らかである。この同時制御の結果として、ヘモグロビン濃度の変動が、体積およびヘモグロビン含有量の変動よりも小さくなる（8）。

20

【0032】

本明細書で開示されるパラメータは、TTおよびIDA患者のRBCが、正常状態にあるときよりも、絶対的および相対的に、はるかに小さい体積および低いヘモグロビン含有量で末梢に残ることを示す。この残存は、このような貧血の非効率的な赤血球生成に反応したクリアランスの代償性遅延を反映し得る。したがって、RBCクリアランスの要因となる挙動を変化させ得る機構が存在しなければならない。TTおよびIDAのRBCクリアランスを健常者またはACD患者のものと比較すると、要因を特定する新しい手段を提供することができる。v_cの変動が、健常者の

30

【0033】

【数3】

\bar{v}

の変動よりもはるかに小さく（14）、これは要因がMCHC線上の位置と非常に相関していることを提示する。

【0034】

貧血

貧血は、血流中に存在するヘモグロビンの量が正常よりも少ない状態である。貧血を、赤血球の大きさ（MCVまたは平均赤血球体積として測定される）に基づき、主に3つの群に細分することができる。正球性貧血は、赤血球が正常な大きさ（すなわち、80～100fL）であるときに診断される。大球性貧血（通常、B12欠乏により生じる）は、赤血球の大きさが正常よりも大きいとき、すなわち、MCV > 100fLであるときに診断される。小球性貧血は、赤血球が正常よりも小さいとき、すなわち、MCV < 80fLのときである。

40

【0035】

貧血予備軍の診断 / 貧血発症の予測

本明細書で示すモデルおよび方法は、クリアランス遅延の兆候を探ることにより、明らかな臨床的貧血が発症する前に、潜伏貧血または代償性貧血の患者を特定するための可能な方法を提供する。この可能性は、正常CBCを持っていたが、少なくとも30日および

50

90日未満後に別の正常CBCまたは臨床的なIDAが生じた患者の独立集合において検査された。各患者のサンプルについて、全赤血球の(v、h)座標がMCHC線上に投影され、確率密度が、平均($P_{0.85}$)の85%未満でこの線に沿って統合された。図3A~Cは、1患者のCBCが正常から潜伏IDAおよび最終的にIDAへ進む様子を示す。中央パネル(図3B)に示すCBCは、臨床的には目立たないものであるが、 $P_{0.85}$ が異常であり、発症してはいないが51日の間に治療することになる貧血を予測している。図3Dは、健康なままである患者の20の正常CBCと30~90日後にIDAを発症した患者の20の正常CBCについての $P_{0.85}$ の値を示す。 $P_{0.85}$ の値は、75%の感度および100%の特異性でIDAを予測した。実施例3「鉄欠乏性貧血の予測」を参照されたい。現在の標準治療は、すべてのCBCが「正常」であるため、この母集団で0%の感度を有する。このモデルに基づく予測は、異なる源および異なる時間点からの情報の複数の測定および種類の統合に依拠することの多い統計的回帰手法と対照的に、一時の単一CBC測定のみで依拠する(15)。

10

【0036】

したがって、説明した方法を使用して、貧血予備軍を診断することができ、または発症の少なくとも30~90日前、例えば、被験者がまだ正常なCBCを有する(例えば、被験者のHCT、MCV、RDW、MCH、MCHC、RBC、およびHGBレベルがすべて正常範囲内にある)ときに、被験者の貧血の発症を予測することができる。

【0037】

正常CBC値は、分析を行う研究所によって変化し得るが、一般に、正常値は以下のとおりである。

20

【0038】

【表1】

パラメータ	値
ヘマトクリット(HCT) (高度によって変化する)	男性: 40.7~50.3% 女性: 36.1~44.3%
平均赤血球容積(MCV)	80~95 フェムトリットル
赤血球分布幅(RDW)	11.5~14.5%
平均赤血球ヘモグロビン (MCH) (MCHC)	27~31 pg/細胞 32~36 gm/dL
赤血球(RBC)数 (高度によって変化する)	男性: 470~610万 細胞/uL 女性: 420~540万 細胞/uL
ヘモグロビン(HGB) (高度によって変化する)	男性: 13.8~17.2 gm/dL 女性: 12.1~15.1 gm/dL

30

【0039】

方法は、被験者のサンプル中の各赤血球について、赤血球体積およびHGB含有量(例えば、HGB質量または濃度)を判定するステップと、サンプル中の全赤血球にわたる体積およびHGB含有量値の分布を計算するステップと、例えば図5に示すような各赤血球体積およびヘモグロビンの変換を計算するステップと、サンプル中の全赤血球についてのこのような変換の分布を計算するステップと、例えば、図3A~Dおよび8A~Dに示すような、変換された体積およびHGB含有量の閾値レベルより変換が低い赤血球の百分率を判定するステップとを含む。一部の実施形態では、方法が、例えば、被験者の赤血球を含むサンプル中で、母平均赤血球ヘモグロビン濃度(MCHC)を判定するステップと、サンプル中の全赤血球にわたる体積およびHGB含有量値の分布を計算するステップと、例えば図5に示すような各赤血球体積およびヘモグロビンの変換を計算するステップと、サンプル中の全赤血球についてのこのような変換の分布を計算するステップと、変換が平均変換の閾値百分率よりも低いサンプル中の赤血球分画を判定して、サンプル分画を提供するステップと、サンプル分画を基準分画と比較するステップとを含み、基準分画よりも低いサンプル分画があれば、被験者が貧血、例えばIDAを発症する危険があることを示

40

50

す。閾値百分率および基準レベルの値を、当技術で公知の方法を使用して選択することができ、この値は感度および特異性を最大化する値を表すことができる。一部の実施形態では、閾値百分率が母平均MCHCの70%、75%、80%、85%、90%、または95%である。一部の実施形態では、基準分画が約0.10、0.11、0.12、0.13、0.14、または0.15である。当業者は、最適な閾値百分率および基準分画を、公知の統計的な方法を使用して容易に特定することができるだろう。

【0040】

鑑別診断 サラセミア対鉄欠乏性貧血

小球性貧血の鑑別診断は、鉄欠乏性貧血（IDA）、サラセミア形質（TT）慢性疾患に伴う貧血（ACD）、および他の原因による貧血を含む。

10

【0041】

ACDは、通常、正常値の下限よりも20%未満までのヘマトクリットの減少と、正常または高いフェリチンと、低いまたは正常な総鉄結合能とを伴う。減少したRBCクリアランス閾値（ v_c ）が、貧血を補う適応生理学反応を表す場合には、ACDについてここで見られる正常 v_c がおそらく予想され、貧血自体が適応生理学反応を表し得る（18）。慢性疾患を有する被験者に貧血があれば、通常、ACDの診断を提案する。通常ACDと関連する慢性疾患として、自己免疫疾患、例えば、クローン病、全身性エリテマトーデス、関節リウマチ、および潰瘍性大腸炎；癌、例えば、リンパ腫およびホジキン病；慢性腎臓病；肝臓病、例えば、肝硬変；慢性感染症、例えば、細菌性心内膜炎、骨髄炎（骨感染症）、HIV/AIDS、B型肝炎、またはC型肝炎がある。例えば、GardnerおよびBenz Jr.、「Anemia of chronic diseases.」 In: Hoffman他編、Hematology: Basic Principles and Practice.、第5版、Philadelphia, Pa: Elsevier Churchill Livingstone、2008、37章を参照されたい。

20

【0042】

IDAは通常、ヘマトクリットの減少、低いMCV、および低いフェリチンを伴う。軽度のIDAは通常、正常値の下限よりも20%未満までのヘマトクリットの減少を伴う。IDAの患者は、正常ヘマトクリットを含む正常MCVの既存の証拠を示し得る。

【0043】

TTは通常、高ヘモグロビンA2分画、または1もしくは複数のアルファグロビン遺伝子突然変異の存在と関連する。TTは、通常、正常値の下限よりも20%未満までのヘマトクリットの減少と、低いMCV、および正常なフェリチンを伴うことが多い。TTは、世界中で最も一般にスクリーニングされている疾患の1つであるが、既存の診断方法は非常に高価であるか、受け入れがたいほど診断精度が低く、偽陽性率が30%にのぼる（19、20）。本明細書に記載のモデルおよび方法は、IDAとTTと、場合によっては他の原因の貧血とを区別する、新しく、おそらくはより正確な方法を提供する。

30

【0044】

図2は、 D_h が、小球性貧血の最も一般的な2つの原因であるIDAとTTとを区別することを示す。10の訓練サンプルを分析することにより、 D_h の例示的な閾値値0.0045が設定された。その後、軽度IDAまたはTTの診断を明確に行うことのできる50の独立した患者サンプルが分析され、 D_h が計算された。図4は、この D_h 閾値が98%の診断精度を有しており、IDAの22ケースのうち22ケース、およびTTの28ケースのうち27ケースを正確に特定し、他の公開された手法よりも6~41%優れていることを示す（20）。実施例4「小球性貧血の鑑別診断」を参照されたい。

40

【0045】

したがって、本明細書に記載の方法は、小球性貧血の被験者において、鉄欠乏性貧血（IDA）とサラセミア形質（TT）と、場合によっては他の原因の貧血とを、被験者のサンプル中の平均ヘモグロビン含有量（ D_h ）の変動の大きさの値を判定するステップと、 D_h を基準値と比較するステップとを含む方法により、鑑別診断するステップを含む。基準値よりも高い D_h があれば、被験者がIDAであることを示し、基準値よりも低い D_h があれば、被験者がTTであることを示す。

50

【 0 0 4 6 】

赤血球生成促進剤（例えば、エリスロポエチン（EPO））および/または鉄補給剤療法の最適化

本明細書に記載の方法を使用して、EPOまたは鉄補給剤療法を最適化することができる。例えば、方法は、EPO治療を受けている被験者のサンプル中の正規化臨界体積、 v_c を判定するステップを含み得る。

【 0 0 4 7 】

一部の実施形態では、EPOまたは鉄補給剤の投与量を調節して、 v_c または D_h を所望レベル、例えば、閾値レベルよりも高いレベルもしくは低いレベル、または所望範囲内のレベルで維持する。例えば、本明細書で言及するように、正常な健常者の v_c は母平均体積

10

【 0 0 4 8 】

【数4】

 \bar{v}

の約80%、または、一般的な90fLのMCVについて約72fLである。したがって、一部の実施形態では、方法が、EPOまたは鉄補給剤の投与量を調節して、被験者の

【 0 0 4 9 】

【数5】

 \bar{v}

20

の70%よりも高い v_c 、被験者の

【 0 0 5 0 】

【数6】

 \bar{v}

の75%よりも高い v_c 、被験者の

【 0 0 5 1 】

【数7】

 \bar{v}

30

の75~85%よりも高い v_c を維持するステップを含む。一部の実施形態では、方法が、治療を調節して70%よりも高い v_c または70~75%の v_c を維持するステップを含む。一部の実施形態では、 v_c または D_h を時間とともに監視して、被験者に対するEPOまたは鉄補給剤の投与量を調節して、 v_c または D_h を選択された閾値よりも高くもしくは低く、または所与の範囲内で維持する。

【 0 0 5 2 】

G I疾患の危険性が高い被験者の特定

IDAまたは他の原因不明の貧血は、結腸癌（16）等の消化器疾患および小児期の栄養不良（17）を含む深刻な疾患の最初の発現であることが多い。IDAに関連する他のG I疾患として、結腸直腸（例えば、結腸）癌、消化管潰瘍（胃潰瘍、消化性潰瘍、盲腸潰瘍）、憩室炎、腸管虚血、胃癌、胃炎、食道炎、G Iポリープ、炎症性腸疾患（クローン病、潰瘍性大腸炎）、およびセリアック病がある。この方法により貧血を初期に検出または予測することによって、このような疾患により早期に対応することができる。したがって、例えば、G I疾患（例えば、大腸内視鏡検査を使用して）または栄養評価について、スクリーニングまたは勧告追加スクリーニングのために、本明細書の方法を使用することができる。適切な患者の精密検査、例えば大腸内視鏡検査をより早期に開始することができるようになり、また適切な治療、例えば鉄補給剤の処方より早期に開始することができるようになる。

40

【 0 0 5 3 】

50

血液ドーピングの検出

特に有酸素運動における運動選手のパフォーマンスは、存在するRBCの数によって影響され得る。RBCが多いほど、肺から運動中の筋肉への酸素の運搬および供給能力が大きくなる。赤血球の一時的な急増は、血液ドーピングと呼ばれる輸血、またはエリスロポエチン、類似体、および模倣薬等の赤血球生成促進剤の使用によって得られる。血液ドーピングは、検出が困難なことで知られている。例えば、JelkmannおよびLundby、Blood、2011年9月1日、118(9)、2395-404、Segura他、「Current strategic approaches for the detection of blood doping practices」、Forensic Sci Int. 2011年8月31日、電子ジャーナルを参照されたい。本方法を使用して、血液ドーピングまたは赤血球生成促進剤の使用の可能性を示す、RBC体積およびHGB含有量の異常分布の存在を検出することができる。例えば、一部の実施形態では、平均ヘモグロビン含有量(D_h)の変動の大きさ、および平均赤血球体積(D_v)の変動の大きさの一方または両方の値を、血液ドーピングまたは赤血球生成促進剤の使用の疑いのある被験者のサンプルにおいて判定し、基準値よりも高い D_h および/または D_v がある場合、血液ドーピングの存在もしくは使用または赤血球生成促進剤の使用を示す。

10

【0054】

サンプル、システム、ソフトウェア、および試験方法

本明細書に記載の方法は、RBCを損なうことなく保存する公知の採取方法(例えば、適量の真空(拔出し)と、RBCを実質的な溶血なしに採取可能な十分な大きさの針、例えば、少なくとも25g以上の針とを使用する採血)を用いて得られた末梢血液サンプルを使用して実施される。採取後24時間、12時間、または6時間以内に測定が行われることが好ましい。RBC体積(例えば、低角度(2° ~ 3°)の散乱検出を使用)とヘモグロビン質量または濃度(例えば、高角度(5° ~ 15°)の散乱検出を使用)の両方を測定可能な当技術で公知の方法または装置を使用して、網状赤血球およびCBCの測定を行うことができる。例示的な方法は、米国特許出願公開第2011/0,178,716号、米国特許出願公開第2011/0,164,803号、米国特許出願公開第2011/0,149,061号、米国特許出願公開第2011/0,077,871号、米国特許出願公開第2011/0,070,606号、および米国特許出願公開第2011/0,070,210号に記載されている。

20

【0055】

一部の実施形態では、血液分析器、例えば、手動、半自動、または自動の血液分析器を使用して測定が行われる。血液分析器の例は、当技術で公知であり、例えば、参照により本明細書に組み込まれている、米国特許第5,017,497号、米国特許第5,266,269号、米国特許第5,378,633号、米国特許第5,631,165号、米国特許第5,812,419号、米国特許第6,228,652号、米国特許第6,524,858号、米国特許第6,320,656号、米国特許第7,324,194号、米国特許第7,981,681号、米国特許出願公開第2008/0,153,170号、米国特許出願公開第US2008/0,158,561号、米国特許出願公開第S2008/0,268,494号、米国特許出願公開第US2011/0,178,716号、米国特許出願公開第2011/0,077,871号、および米国特許出願公開第2011/0,070,606号に記載されている。本方法で有用な血液分析器は、当技術で公知のあらゆる検出方法、例えば、フローサイトメトリー、光学もしくは画像ベース分析またはインピーダンススペースを使用することができる。本方法で有用な血液分析器は、通常、CBCの全パラメータを測定可能なものである。すなわち、分析器は、少なくとも赤血球(RBC)の赤血球体積(CV)と、血球ヘモグロビン濃度(CHC)または血球ヘモグロビン質量(CH)とを判定することができるべきである。

30

40

【0056】

血液分析器のいくつかのモデルが、例えば、Abbott Laboratories (Abbott Park, IL, United States) (例えばCell-Dyn Sapphire)、およびSiemens (Deerfield, IL, Uni

50

ted States) (例えば、Advia 120または2120自動血液分析器) から市販されている。他の製造業者として、Beckman Coulter、Inc. (Fullerton, CA, United States)、東亜医用電子(株)(日本、神戸)、Constitution Medical (Boston, MA)、およびHORIZON ABX Inc (Irvine, CA, United States) がある。

【0057】

本発明はまた、臨床的なサンプルを測定するための検出モジュールと、コンピューティング装置とを備えた血液分析器システムを提供する。このコンピューティング装置は、検出モジュールと通信し、(i) 母平均赤血球ヘモグロビン濃度(MCHC)、ならびに、サンプル分画を判定するための、変換された体積およびヘモグロビン含有量がすべての変換された体積およびヘモグロビン含有量の平均の閾値百分率よりも低いサンプル中の赤血球分画、(ii) ヘモグロビン含有量減少率(D_h)の母集団内赤血球間での変動の大きさ、(iii) 赤血球体積減少率(D_v)の母集団内赤血球間での変動の大きさ、および/または(iv) 正規化臨界体積(v_c)もしくは他のクリアランス閾値の1または複数を、検出モジュールの出力に基づいて判定するためのプログラミングを備える。一部の実施形態では、コンピューティング装置が、血液分析器からの入力を使用する別個のコンピュータである。一部の実施形態では、コンピューティング装置が血液分析器に組み込まれるか、または血液分析器の一部となっている。

【0058】

一般に、検出モジュールは、分析のために赤血球、例えば全血を含むサンプルを保持するように構成された分析チャンバを備える。一部の実施形態では、検出モジュールが、サンプルを分析するように構成されたフローサイトメータを備える。一部の実施形態では、検出モジュールが、サンプルを分析するように構成された光学または画像分析器を備える。

【0059】

本明細書では、(i) 母平均赤血球ヘモグロビン濃度(MCHC)、ならびに、サンプル分画を判定するための、変換された体積およびヘモグロビン含有量がすべての変換された体積およびヘモグロビン含有量の平均の閾値百分率よりも低いサンプル中の赤血球分画、(ii) ヘモグロビン含有量減少率(D_h)の母集団内赤血球間での変動の大きさ、(iii) 赤血球体積減少率(D_v)の母集団内赤血球間での変動の大きさ、および/または(iv) 正規化臨界体積(v_c)もしくは他のクリアランス閾値の1または複数を計算するためのプログラミングを備えたコンピュータ読取可能な媒体も提供される。

【0060】

一部の実施形態では、血液分析器からのデータ(例えば、赤血球(RBC)の赤血球体積(CV)、平均赤血球体積(MCV)、血球ヘモグロビン濃度(CHC)または血球ヘモグロビン含有量(CH)、平均赤血球ヘモグロビン濃度(MCHC)、および平均赤血球ヘモグロビン含有量(MCH)、ならびにこれらの母集団統計)がコンピューティング装置によって受信され、コンピューティング装置は、(i) 母平均赤血球ヘモグロビン濃度(MCHC)、ならびに、サンプル分画を判定するための、変換された体積およびヘモグロビン含有量がすべての変換された体積およびヘモグロビン含有量の平均の閾値百分率よりも低いサンプル中の赤血球分画、(ii) ヘモグロビン含有量減少率(D_h)の母集団内赤血球間での変動の大きさ、(iii) 赤血球体積減少率(D_v)の母集団内赤血球間での変動の大きさ、および/または(iv) 正規化臨界体積(v_c)もしくは他のクリアランス閾値の1または複数を計算するために、本明細書に記載のアルゴリズムを含むプログラミングを実行する。

【0061】

一部の実施形態では、コンピューティング装置が血液分析器からの入力を使用する別個のコンピュータである。一部の実施形態では、コンピューティング装置が血液分析器に組み込まれるか、または血液分析器の一部となっている。

【 0 0 6 2 】

プログラミングは、物理的記憶装置または伝送媒体において提供され得る。命令を受信するコンピューティング装置（例えば、血液分析器システムの一部である別個の装置または情報処理モジュール）は、アルゴリズムを実行し、かつ/または本方法から入手したデータを処理する。コンピュータ読取可能な記憶媒体の例として、コンピュータ内部または外部にあってもなくても、フロッピーディスク、磁気テープ、CD-ROM、ハードディスクドライブ、ROMもしくは集積回路、光磁気ディスク、またはPCMCIAカード等のコンピュータ読取可能なカード等がある。ファイルに含まれる情報を、コンピュータ読取可能な媒体に記憶することができ、「記憶」とは、後日、ローカルまたは遠隔ネットワーク上のコンピュータによりアクセス可能および検索可能となるように情報を記録することを意味する。一部の実施形態では、本明細書に記載の方法が、サンプルが実施される（run）度に自動的に実行される。

10

【 0 0 6 3 】

図14は、本明細書に記載した技術とともに使用可能なコンピュータ装置700および携帯型コンピュータ装置750の例を示す。コンピューティング装置700は、例えば、ラップトップ、デスクトップ、ワークステーション、パーソナルデジタルアシスタント、サーバ、ブレードサーバ、メインフレーム、およびその他の適切なコンピュータを含む種々の形のデジタルコンピュータを表すものとする。コンピューティング装置750は、例えば、パーソナルデジタルアシスタント、携帯電話、スマートフォン、およびその他の同様のコンピューティング装置を含む種々の形の携帯型装置を表すものとする。ここに示す構成要素、それらの接続および関係、ならびに機能は例にすぎず、本書類に記載し、かつ/または特許請求した技術の実施を限定するものではない。

20

【 0 0 6 4 】

コンピューティング装置700は、ラップトップ、デスクトップ、ワークステーション、パーソナルデジタルアシスタント、サーバ、ブレードサーバ、メインフレーム、および、血液分析器システムまたは装置に組み込まれるコンピュータを含むその他の適切なコンピュータを含む種々の形のデジタルコンピュータを表すものとする。コンピューティング装置750は、例えば、パーソナルデジタルアシスタント、携帯電話、スマートフォン、およびその他の同様のコンピューティング装置を含む種々の形の携帯型装置を表すものとする。ここに示す構成要素、それらの接続および関係、ならびに機能は例にすぎず、本書類に記載し、かつ/または特許請求した技術の実施を限定するものではない。

30

【 0 0 6 5 】

コンピューティング装置700は、プロセッサ702、メモリ704、記憶装置706、メモリ704および高速拡張ポート710に接続する高速ユーザインターフェース708、ならびに低速バス714および記憶装置706に接続する低速ユーザインターフェース712を備える。構成要素702、704、706、708、710、712のそれぞれは、種々のバスを使用して相互接続され、共通のマザーボードに取り付けられるか、または必要に応じて他の方法で取り付けられ得る。プロセッサ702は、コンピューティング装置700内で実行するための命令を処理することができ、この命令には、メモリ704または記憶装置706に記憶された、例えば、高速ユーザインターフェース708に結合されたディスプレイ716を含む外部入出力装置にGUIのための図形情報を表示する旨の命令が含まれる。他の実施では、複数のメモリおよび複数のタイプのメモリとともに、複数のプロセッサおよび/または複数のバスを必要に応じて使用することができる。また、複数のコンピューティング装置700を接続することができ、各装置が必要な動作の一部を提供する（例えば、サーババンク、ブレードサーバ群、またはマルチプロセッサシステムとして）。

40

【 0 0 6 6 】

メモリ704は、コンピューティング装置700内で情報を記憶する。一実施では、メモリ704が1または複数の揮発性メモリユニットである。別の実施では、メモリ704が1または複数の不揮発性メモリユニットである。メモリ704は、例えば、磁気ディス

50

くまたは光ディスクを含む別の形のコンピュータ読取可能な媒体であってもよい。

【0067】

記憶装置706は、コンピューティング装置700に大容量記憶を提供することができる。一実施では、記憶装置706がコンピュータ読取可能な媒体であるか、またはコンピュータ読取可能な媒体を含むことができる。コンピュータ読取可能な媒体としては、例えば、フロッピーディスク装置、ハードディスク装置、光ディスク装置、テープ装置、フラッシュメモリもしくは他の同様の固体メモリ装置、またはストレージエリアネットワーク内の装置または他の構成を含む多くの装置がある。コンピュータプログラム製品を、情報担体で明確に具体化することができる。コンピュータプログラム製品は、実行時に、例えば前述した方法を含む1または複数の方法を実施する命令を含むことができる。情報担体は、例えば、メモリ704、記憶装置706、プロセッサ702のメモリ等を含む、コンピュータまたは機械読取可能な媒体である。

10

【0068】

高速コントローラ708は、コンピューティング装置700の帯域幅集約動作を管理し、低速コントローラ712は低帯域幅集約動作を管理する。このような機能の割当ては一例にすぎない。一実施では、高速コントローラ708がメモリ704、ディスプレイ716（例えば、図形プロセッサまたはアクセラレータを通して）、および種々の拡張カード（図示せず）を受けることのできる高速拡張ポート710に結合される。この実施では、低速コントローラ712が記憶装置706および低速拡張ポート714に結合される。種々の通信ポート（例えば、USB、Bluetooth（登録商標）、イーサネット、無線イーサネット）を含み得る低速拡張ポートを、例えば、キーボード、ポインティングデバイス、スキャナ、または例えばネットワークアダプタを通した、例えばスイッチもしくはルータを含むネットワークデバイスを含む1または複数の入出力装置に結合することができる。

20

【0069】

コンピューティング装置700は、図示するようにいくつかの異なる形で実施可能である。例えば、コンピューティング装置700を、標準サーバ720として、またはこのようなサーバ群において複数回実施することができる。また、コンピューティング装置700をラックサーバシステム724の一部として実施することもできる。加えて、または代わりに、コンピューティング装置700を、例えば、ラップトップコンピュータ722を含むパーソナルコンピュータ内で実施することができる。一部の例では、コンピューティング装置700の構成要素を、例えば、装置750を含む携帯型装置（図示せず）の他の構成要素と組み合わせることができる。このような装置はそれぞれ、コンピューティング装置700、750の1または複数を含むことができ、システム全体を、互いに通信する複数のコンピューティング装置700、750から構成することができる。

30

【0070】

コンピューティング装置750は、他の構成要素のうち、プロセッサ752、メモリ764、ならびに例えば、ディスプレイ754、通信ユーザインターフェース766、およびトランシーバ768を含む入出力装置を備える。装置750は、例えば、マイクロドライブまたは他の装置を含む記憶装置を備えて、追加の記憶容量を設けることができる。構成要素750、752、764、754、766、768のそれぞれは、種々のバスを使用して接続され、構成要素のうちの複数を共通のマザーボードに取り付けるか、または必要に応じて他の方法で取り付けることができる。

40

【0071】

プロセッサ752は、メモリ764に記憶された命令を含む、コンピューティング装置750内の命令を実行することができる。プロセッサを、別個の複数のアナログおよびデジタルプロセッサを備えるチップのチップセットとして実施することができる。プロセッサは、例えば、ユーザインターフェースの制御、装置750により実行されるアプリケーション、および装置750による無線通信を含む、例えば装置750の他の構成要素の調整を行うことができる。

50

【0072】

プロセッサ752は、制御ユーザインターフェース758およびディスプレイ754に結合されたディスプレイユーザインターフェース756を通してユーザと通信することができる。ディスプレイ754は、例えば、TFT LCD（薄膜トランジスタ液晶ディスプレイ）もしくはOLED（有機発光ダイオード）ディスプレイ、または他の適切なディスプレイ技術とすることができる。ディスプレイユーザインターフェース756は、ディスプレイ754を駆動して図形および他の情報をユーザに提示するための適切な回路を備えることができる。制御ユーザインターフェース758は、ユーザからのコマンドを受け、このコマンドをプロセッサ752へ送信するために変換する。加えて、外部ユーザインターフェース762は、装置750と他の装置との近領域通信を可能にするように、プロセッサ742と通信することができる。外部ユーザインターフェース762は、例えば、一部の実施における無線通信、または他の実施における無線通信を行うことができ、複数のユーザインターフェースを使用することもできる。

10

【0073】

メモリ764は、コンピューティング装置750内の情報を記憶する。メモリ764を、1もしくは複数のコンピュータ読取可能な媒体、1もしくは複数の揮発性メモリユニット、1もしくは複数の不揮発性メモリユニットの1または複数として実施することができる。拡張メモリ774を設けて、例えば、SIMM（シングルインラインメモリモジュール）カードユーザインターフェースを含み得る拡張ユーザインターフェース772を通して装置750に接続することもできる。このような拡張メモリ774は、装置750のための追加の記憶空間を提供することができ、または装置750のためのアプリケーションもしくは他の情報を記憶することもできる。すなわち、拡張メモリ774は、前述したプロセスを実施または補う旨の命令を含むことができ、確実な情報を含むこともできる。したがって、例えば、拡張メモリ774を装置750のセキュリティモジュールとして設けることができ、装置750の確実な使用を可能にする命令を、拡張メモリ774にプログラムすることができる。加えて、例えば、特定情報をSIMMカード上にハッキング不可能に配置することを含む追加の情報とともに、SIMMカードを通して確実なアプリケーションを提供することができる。

20

【0074】

メモリは、例えば、後述するようにフラッシュメモリおよび/またはNVRAMメモリを含むことができる。一実施では、コンピュータプログラム製品が、情報担体で明確に具体化される。コンピュータプログラム製品は、実行時に、例えば前述した方法を含む1または複数の方法を実施する命令を含むことができる。情報担体は、例えば、トランシーバ768または外部ユーザインターフェース762によって受信可能な、例えば、メモリ764、拡張メモリ774、および/またはプロセッサ752のメモリを含む、コンピュータまたは機械読取可能な媒体である。

30

【0075】

装置750は、必要な場合にデジタル信号処理回路を含み得る通信ユーザインターフェース766を通して無線通信することができる。通信ユーザインターフェース766は、例えば、特にGSM音声通話、SMS、EMS、もしくはMMSメッセージ、CDMA、TDMA、PDC、WCDMA、CDMA2000、またはGPRSを含む種々のモードまたはプロトコルによる通信を行うことができる。このような通信は、例えば、高周波トランシーバ768を通して行うことができる。加えて、例えば、Bluetooth（登録商標）、WiFi、または他のこのようなトランシーバ（図示せず）の使用を含む短距離通信を行うことができる。加えて、GPS（全地球測位システム）レシーバモジュール770は、装置750で実行されているアプリケーションにより必要に応じて使用可能な追加のナビゲーション関連および位置関連無線データを装置750に提供することができる。

40

【0076】

装置750は、ユーザから話される情報を受信して、使用可能なデジタル情報に変換す

50

ることのできる音声コーデック760を使用して音声通信することができる。音声コーデック760は、同様に、例えば、装置750の送受信器内のスピーカを通して、ユーザに対して可聴音を発生することができる。このような音は、音声通話からの音を含むことができ、録音された音（例えば、音声メッセージ、音楽ファイル等）を含むことができ、また装置750で動作するアプリケーションにより発生する音を含むことができる。

【0077】

コンピューティング装置750を、図示したように、複数の異なる形で実施することができる。例えば、コンピューティング装置750を携帯電話780として実施することができる。コンピューティング装置750を、スマートフォン782、パーソナルデジタルアシスタント、またはその他の同様の携帯型装置の一部として実施することもできる。

10

【0078】

本明細書に記載のシステムおよび技術の種々の実施を、デジタル電子回路、集積回路、特別に設計されたASIC（特定用途向け集積回路）、コンピュータハードウェア、ファームウェア、ソフトウェア、および/またはこれらの組合せにおいて実現することができる。これらの種々の実施は、少なくとも1つのプログラム可能プロセッサを含むプログラム可能システムで実行可能かつ/または解釈可能な1または複数のコンピュータプログラムにおける実施を含むことができる。このシステムは、専用または汎用とすることができ、記憶システム、少なくとも1つの入力装置、および少なくとも1つの出力装置からデータおよび命令を受信し、かつこれらに対してデータおよび命令を送信するように結合される。

20

【0079】

このようなコンピュータプログラム（プログラム、ソフトウェア、ソフトウェアアプリケーション、またはコードとしても知られる）は、プログラム可能プロセッサのための機械命令を含み、高レベル手続き型および/またはオブジェクト指向プログラミング言語、ならびに/またはアセンブリ/機械言語において実施可能である。本明細書で使用されるように、機械読取可能な媒体およびコンピュータ読取可能な媒体という用語は、機械命令を受信する機械読取可能な媒体を含むプログラム可能なプロセッサに機械命令および/またはデータを与えるために使用されるコンピュータプログラム製品、装置および/またはデバイス（例えば、磁気ディスク、光ディスク、メモリ、プログラム可能論理回路（PLD））を指す。

30

【0080】

ユーザとの対話を行うために、本明細書に記載のシステムおよび技術を、情報をユーザに表示するためのディスプレイ装置（例えば、CRT（陰極線管）またはLCD（液晶ディスプレイ）モニタ）、ユーザがコンピュータに入力可能なキーボードおよびポインティングデバイス（例えば、マウスまたはトラックボール）を有するコンピュータで実施することができる。他の種類の装置を使用して、ユーザとの対話を行ってもよい。例えば、ユーザに与えられるフィードバックは、感覚フィードバック（例えば、視覚フィードバック、聴覚フィードバック、もしくは触覚フィードバック）の形とすることができ、ユーザからの入力を、音響、音声、または触覚入力を含む形で受信することができる。

【0081】

40

本明細書に記載のシステムおよび技術を、演算システムで実施することができ、この演算システムは、バックエンドコンポーネント（例えば、データサーバとして）、ミドルウェアコンポーネント（例えば、アプリケーションサーバ）、フロントエンドコンポーネント（例えば、本明細書に記載のシステムおよび技術の実施とユーザが対話する際に通す、グラフィカルユーザインターフェースまたはウェブブラウザを有するクライアントコンピュータ）、またはこのようなバックエンドコンポーネント、ミドルウェアコンポーネント、もしくはフロントエンドコンポーネントの組合せを含む。システムのコンポーネントを、デジタルデータ通信（例えば、通信ネットワーク）の形または媒体により相互接続することができる。通信ネットワークの例として、ローカルエリアネットワーク（LAN）、広域ネットワーク（WAN）、およびインターネットがある。

50

【 0 0 8 2 】

演算システムは、クライアントおよびサーバを含むことができる。クライアントおよびサーバは、一般に互いにリモートであり、通常は通信ネットワークを通じて対話する。クライアントおよびサーバの関係は、各コンピュータで実行され、互いにクライアント - サーバ関係を有するコンピュータプログラムによって発生する。

【 0 0 8 3 】

一部の実施では、本明細書に記載のエンジンが、単一または複合エンジンと別個であっても、組み合わされていても、組み込まれていてもよい。図示したエンジンは、本明細書に記載のシステムを、図示したソフトウェアアーキテクチャに限定するものではない。

【 0 0 8 4 】

実施例

本発明をさらに以下の実施例で説明する。この実施例は、特許請求の範囲に記載された本発明の範囲を限定するものではない。

【実施例 1】

【 0 0 8 5 】

モデル開発

生体内の個々の R B C の寿命中における体積およびヘモグロビン制御は非常に複雑で、解明が困難である。R B C の大きな母集団の平均挙動を解明するほうが扱いやすい。この母集団レベル挙動を見抜くために、R B C 母集団の動力学を説明する R B C 成熟およびクリアランスのモデルが開発された。モデルは、時間 (t) にわたる平均 R B C の体積 (v) およびヘモグロビン (h) 動力学を、決定論的還元 (f) と、特定の関数形式が変動し得る不規則変動 () とに分解する。式 1 に一例を示し、式中、v および h は母平均

【 0 0 8 6 】

【数 8】

$$(\bar{v}, \bar{h})$$

によってスケールリングされ、t は平均赤血球齢

【 0 0 8 7 】

【数 9】

$$(\bar{\tau})$$

によってスケールリングされる。先のレポート (3 ~ 5 、 7) からのデータに基づき、2 つのパラメータが、決定論的コンポーネント、すなわち R B C が M C H C 線に近くなるまでその効果が優位である高速変化 () と、低速変化 () とに挿入された。図 1 B および式 1 に示すように、平均ゼロおよび分散が拡散テンソル 2 D により与えられた、ガウスまたは同様の分布を持つ不規則変数として、不規則変動をモデリングすることができる。

【 0 0 8 8 】

10

20

30

【数 1 0】

$$\begin{bmatrix} \frac{dv}{dt} \\ \frac{dh}{dt} \end{bmatrix} = \mathbf{f} + \zeta$$

(1)

$$\mathbf{f} = \begin{cases} \alpha \cdot e^{\beta_v(v-h)} \\ \alpha \cdot e^{\beta_h(h-v)} \end{cases}$$

10

$$\zeta = \begin{cases} N(0, 2D_v) \\ N(0, 2D_h) \end{cases}$$

【0089】

一般的な逆問題および特別なヒト病態生理学(9)のように、この問題は、fの異なる関数形式が生体内動力学を再現するという意味で不良設定である。fの正確な関数形式は重要ではない。このモデルの挙動は高速および低速の決定論的動力学および不規則変動の定性的組合せに依拠した。表1A~1Bは、モデルで使用可能な異なるタイプの関数形式を示す。

20

【0090】

【表 2】

表1A. fの関数形式

	f
A	$f_v = -\alpha \cdot e^{\beta_v(v-h)}$ $f_h = -\alpha \cdot e^{\beta_h(h-v)}$
B	$f_v = -\alpha \cdot \max\{\beta_v(v-h), 1\}$ $f_h = -\alpha \cdot \max\{\beta_h(h-v), 1\}$
C	$f_v = -\alpha \cdot v \cdot \max\{\beta_v(v-h), 1\}$ $f_h = -\alpha \cdot h \cdot \max\{\beta_h(h-v), 1\}$
D	$f_v = -\alpha \cdot v \cdot e^{\beta_v(v-h)}$ $f_h = -\alpha \cdot h \cdot e^{\beta_h(h-v)}$
E	$f_v = -\alpha - \max\{\beta_v(v-h), 0\}$ $f_h = -\alpha - \max\{\beta_h(v-h), 0\}$
D	$f_v = -\alpha_v \cdot v \cdot e^{\beta_v(v-h)}$ $f_h = -\alpha_h \cdot h \cdot e^{\beta_h(h-v)}$

10

20

30

【0091】

【表 3】

表1B. dの関数形式

	d
A	$d(v, h) = \frac{1}{1 + e^{\Delta}}$
B	$d(v, h) = \begin{cases} 1 & \Delta \leq 0 \\ 0 & \Delta > 0 \end{cases}$
C	$d(v, h; k) = \frac{1}{1 + e^{k\Delta}}$

40

図9 A ~ Cは、fの形式A ~ Cに対応する速度場を示し、図10 A、Bは、クリアランス関数(d)形式AおよびBを示す。は母集団内の赤血球の平均齢である。の定義を以

50

下に述べる。

【0092】

パラメータが、決定論的進化（ f ）およびクリアランス関数（ d ）の異なる関数形式について推定された。定性的結果はこれらの異なる関数形式について一致しており、これらの結果が生体内病態生理学の特徴を表し、データの過剰適合がないことを表すことを提示する。関数形式の詳細は、表1A～Bに示され、形式A～Cについての推定は、図2、11、および12の箱型図に示される。

【0093】

本モデルでは、典型的な個々の赤血球についての体積およびヘモグロビン含有量の不規則変動および決定論的散逸または還元が、ポテンシャル（10）中のブラウン運動をモデリングするために通常使用されるランジュバン方程式により説明された。これにより、RBCの全循環母集団の動力学を、フォッカー-プランク方程式（10、11）により近似可能な時間依存性体積 - ヘモグロビン確率分布（ $P(v, h, t)$ ）についてのマスタ方程式によって説明することができる。式2は、この体積 - ヘモグロビン分布についての確率密度のドリフト（ f ）、拡散（ D ）、出生（ b ）、および死滅（ d ）を説明する。

【0094】

【数11】

$$\frac{\partial P}{\partial t} = -\nabla \cdot (Pf) + \nabla \cdot (D \cdot \nabla P) + b(v, h, t) - d(v, h, t)P \quad (2)$$

$$D = \begin{bmatrix} D_v & 0 \\ 0 & D_h \end{bmatrix}$$

【0095】

出生死滅プロセスは、常に母集団に加えられ母集団から除去されるRBCの要因となる。健常および軽度の病気の状態では、加えられる赤血球の総数は、除去される赤血球の総数に等しい。

【0096】

【数12】

$$\iint d(v, h)P \, dv \, dh = \iint b(v, h) \, dv \, dh$$

RBC除去の正確な要因および機構は完全には解明されていない（12）が、図1Aに示すような経験的測定により、MCHC線に沿った閾値（ v_c ）があり、大部分のRBCがMCHC線を超えて除去されたことが提示される。

【0097】

経験的RBC分布の観察に基づき、RBCクリアランスの確率が、RBC体積およびヘモグロビン含有量の関数としてモデリングされる。体積 - ヘモグロビン含有量平面における各RBCの位置がMCHC線上に投影され、クリアランス確率（ d ）が様々に定義され、例えば、この線状のこの投影点から閾値 v_c までの距離のシグモイド（図10A）またはステップ（図10B）関数として定義される。図1、5および表1Bを参照されたい。式3は、この関係を定量化する。

【0098】

10

20

30

40

【数 1 3】

$$d(v, h) = \frac{1}{1 + e^{\Delta}} \quad (3)$$

又は

$$d(v, h) = \begin{cases} 1 & \Delta \leq 0 \\ 0 & \Delta > 0 \end{cases}$$

$$\Delta(v, h) = 100 \cdot \frac{\cos(\theta) \sqrt{(v\bar{v})^2 + (h\bar{h})^2} - v_c \sqrt{\bar{h}^2 + \bar{v}^2}}{v_c \sqrt{\bar{h}^2 + \bar{v}^2}}$$

$$\theta = \tan^{-1}\left(\frac{\bar{h}}{\bar{v}}\right) - \tan^{-1}\left(\frac{h\bar{h}}{v\bar{v}}\right)$$

【0099】

CBC測定は人によって異なるが、これらの動力学プロセスが生体内定常状態P に達することを示す健康者(13)については大きく変化しない。ここで

【0100】

【数14】

$$\lim_{t \rightarrow \infty} P(v, h, t) \rightarrow P_{\infty}(v, h), \text{ すなわち } \frac{\partial P_{\infty}}{\partial t} = 0$$

【0101】

所与のパラメータ集合について、病態生理学的範囲外の体積およびヘモグロビン含有量でゼロになる確率の境界条件と、経験的に測定された網状赤血球分布に等しい初期条件とを有する一次空間導関数

【0102】

【数15】

$$\left(\mathbf{J} = \frac{\Delta_k [f \cdot P](v)}{k} + \frac{\Delta_k [f \cdot P](h)}{k} \right)$$

および二次空間導関数

【0103】

【数16】

$$\left(\mathbf{L} = D_v \frac{\delta_k^2 [P](v)}{k^2} + D_h \frac{\delta_k^2 [P](h)}{k^2} \right)$$

についての有限差分近似を使用して、式2を数値的に解決することができる。体積 - ヘモグロビン含有量平面は、一定のメッシュ幅で離散化され、各メッシュ赤血球中の確率密度に等しい変数のベクトル(P)として表される。ここで報告されるシミュレーション結果は、体積軸に沿った1.8 fLおよびヘモグロビン含有量軸に沿った1.8 pgのメッシュ幅により実行された。このメッシュ幅は、経験的体積およびヘモグロビン含有量測定の解析解と比較可能であった。より小さいメッシュ幅(1.2 fLおよび1.2 pg)を使用してすべての結果が確認された。空間導関数の風上有限差分近似を使用して、対流寄与(f)が数値的にモデリングされた。結果として得られる常微分方程式の線形システムが、MATLABode15s積分器を使用して積分され、定常状態(P)に達するまで反復された。

10

20

30

40

50

【 0 1 0 4 】

数値問題 (P) についての定常状態分布は、解析的にも判定された。進化 (J + L) 項およびクリアランス (d) 項の数値近似は線形演算子であり、出生プロセスの積分スケールリングは、平均年齢の 2 倍の逆数に等しい定数である。

【 0 1 0 5 】

【 数 1 7 】

$$\left(\frac{1}{2\bar{\tau}} \right)$$

線形演算子を反転させて、式 4 に示すように、平均赤血球年齢により示される定常状態分布族を生じさせることができる。 10

【 0 1 0 6 】

【 数 1 8 】

$$\frac{d\mathbf{P}_\infty}{dt} = 0 = -\mathbf{J} \cdot \mathbf{P}_\infty + \mathbf{L} \cdot \mathbf{P}_\infty + d \cdot \mathbf{P}_\infty + \mathbf{P}_0 \int \int d(v, h) =$$

$$(-\mathbf{J} + \mathbf{L} + d) \cdot \mathbf{P}_\infty + \mathbf{P}_0 \frac{1}{2\bar{\tau}} \Rightarrow$$

$$\mathbf{P}_\infty = -(-\mathbf{J} + \mathbf{L} + d)^{-1} \mathbf{P}_0 \frac{1}{2\bar{\tau}}$$

(4)

20

解析および有限差分手法により得られた定常状態分布間の差はごくわずかであった。

【 0 1 0 7 】

パラメータ (、 、 D、および v_c) の適切な選択により、本明細書に記載のモデルは、健常者の RBC 母集団の観察された分布を正確に再現する。

【 実施例 2 】

【 0 1 0 8 】

検証

モデルが健常者の RBC 母集団の動力学を貧血患者の RBC 母集団の動力学と区別することができるか否かを検査するために、原因疾患が異なる 3 種類の最もよく見られる貧血、すなわち炎症性疾患である慢性疾患に伴う貧血 (ACD)、遺伝性疾患であるサラセミア形質 (TT)、および栄養性疾患である鉄欠乏性貧血 (IDA) を有する患者について、CBC および網状赤血球測定を行った (14)。RBC 母集団特性が安定して見え、準定常状態仮定が適正な各貧血の軽度のケースが、複数の見かけの健常対照群として選択された。

【 0 1 0 9 】

血液サンプルおよび CBC 結果は、Partners Healthcare Institutional Review Board により認可された研究プロトコルの下で、三次医療成人病院の臨床検査室から得られた。網状赤血球および CBC 測定が、Siemens Advia 2120 自動血液分析器で、採取後 6 時間以内に行われた (21)。

【 0 1 1 0 】

IDA は、正常値の下限よりも 20% 未満までのヘマトクリットの軽度の減少、低い MCV、低いフェリチン、および正常ヘマトクリットを含む正常 MCV の既存の証拠として定義される。急性疾患、急性出血、輸血後 6 か月以内、並行入院、慢性炎症性疾患、または異常ヘモグロビン症の患者は除外された。

40

50

【0111】

TTは、高ヘモグロビンA 2分画、またはアルファグロビン遺伝子突然変異の存在についてのヘテロ接合体として定義されるとともに、正常値の下限よりも20%未満までのヘマトクリットの減少と、低いMCV、および正常なフェリチンとして定義された。急性疾患、急性出血、輸血後6か月以内、並行入院、慢性炎症性疾患、またはさらなる異常ヘモグロビン症の患者は除外された。

【0112】

ACDは、正常値の下限よりも20%未満までのヘマトクリットの減少と、正常または高いフェリチンと、低いまたは正常な総鉄結合能として定義された。急性疾患、急性出血、輸血後6か月以内、並行入院、慢性炎症性疾患、または異常ヘモグロビン症の患者は除外された。

10

【0113】

各患者のサンプルに対して、患者について観察された定常状態を再現する最適なパラメータ集合(、D、および v_c)が特定された。シミュレーションした定常状態分布と測定したCBC分布との最小二乗適合を使用して、最良適合を特定した。同一の患者について繰返し検査を行うことができるが、適合パラメータの変動はCBC測定の解析変動によって説明されることがわかった。

【0114】

勾配または非勾配最適化方法を使用して、パラメータ空間の最適近傍を各患者について特定した。患者の経験的に測定された網状赤血球分布および初期のランダムに選択されたパラメータ集合を、始点として使用した。式4を使用して、これにより生じる定常状態RBC分布を計算した。この計算された分布(P)を、その後、経験分布(P_{CBC})と比較した。パラメータ推定の質が、式5に示すように、離散化分布に対する正規化残差平方和に等しい目的関数を演算することにより定量化された。式中*i*および*j*は、離散化体積-ヘモグロビン平面の赤血球の指標を表す。

20

【0115】

【数19】

$$C(P_{CBC}^{i,j}, P_{\infty}^{i,j}) = \sum_{i,j} \frac{(P_{CBC}^{i,j} - P_{\infty}^{i,j})^2}{P_{CBC}^{i,j}} \quad (5)$$

30

【0116】

一般に、目的関数は、2つの分布間の相違(または類似)の程度の測定値を提供する。目的関数の値が閾値条件を満たすか否かに基づいて、パラメータ値を調節することができる。例えば、目的関数の値が閾値よりも高くなるか、または低くなるまでパラメータ値を調節することができる。目的関数が相違の測定値(例えば平均二乗差、絶対差の和等)を表す場合、目的関数は、パラメータを調節することによって減少することが求められる。あるいは、目的関数が類似の測定値(例えば相関係数、相互情報等)を表す場合、目的関数は、パラメータを調節することによって増加することが求められる。この例では、この目的関数を減少させるために新しいパラメータ値が選択された。MATLABの勾配に基づく(`lsqnonlin`関数)、ならびに非勾配に基づく(`fminsearch`および`patternsearch`関数)最適化アルゴリズムを使用して、最適パラメータを検索した。すべてのパラメータが非負に制限され、均一な初期パラメータ空間を定義して、1000日よりも長い、または5日よりも短い平均赤血球齢を除外した。その後、ラテン超方格サンプリングを使用して、この空間から初期パラメータを選択した。1つの最適化制限が課され、平均赤血球齢が5日よりも長くなるのに十分な小ささに限定し、平均赤血球齢が1000日よりも短くなるのに十分な大きさに限定する。図6は、モデルが、シミュレーションされた定常状態確率分布と測定された定常状態確率分布とを比較することにより、この健常者についての P_{CBC} を正確に再現することを示す。MCHC線

40

50

に沿って投影されると、この患者についての経験分布はモード近くでわずかに高い密度を有し、モードのいずれかの側までの領域でわずかに低い密度を有していた。

【 0 1 1 7 】

図 7 は、1 人の患者に対する 2 0 0 を超える最適化から得られたパラメータの最小の局所解を示す。一部の結果は、軸の範囲よりも高い局所解を有した。すべてのシミュレーション間の最良適合は、全パラメータについての値の小近傍を形成し、パラメータ推定プロセスが、この患者の血液サンプルについての明確に定義された最適近傍に達したことを示す。

【 0 1 1 8 】

健常者および貧血患者についての適合パラメータが図 2 に示され、中央値適合パラメータが表 2 に挙げられる。

10

【 0 1 1 9 】

【表 4】

表2. 無次元および次元適合パラメータの中央値 (必要な場合)

	正常	ACD	TT	IDA
β_v	26	27	14	15
β_h	16	15	5	12
α	0.05 (0.09 fL/d および 0.03 pg/d)	0.05 (0.09 fL/d および 0.03 pg/d)	0.13 (0.20 fL/d および 0.07 pg/d)	0.13 (0.20 fL/d および 0.07 pg/d)
D_v	0.014 (2.3 fL ² /d)	0.015 (2.4 fL ² /d)	0.017 (2.2 fL ² /d)	0.013 (1.7 fL ² /d)
D_h	0.0014 (0.025 pg ² /d)	2.7×10^{-5} (4.9×10^{-4} pg ² /d)	2.7×10^{-15} (3.6×10^{-14} pg ² /d)	0.019 (0.34 pg ² /d)
v_c	0.80 (72 fL)	0.80 (72 fL)	0.74 (59 fL)	0.71 (56 fL)

20

【 0 1 2 0 】

健常者から得た最良適合パラメータと貧血患者から得た最良適合パラメータとの間には明白な差があり、異なる貧血疾患は異なる特性パラメータ集合を有した。例えば、健常者および A C D 患者は、高い v_c および β_h と低い β_v とを有し、すなわち、低速相中よりも高速相中に体積およびヘモグロビンが比較的多く失われた。これに対して、T T および I D A 患者は、高速相中よりも低速相中に体積およびヘモグロビンが比較的多く失われた。A C D 患者は、健常者に対してわずかに高い D_v とわずかに低い D_h とを示したが、T T は D_h の実質的な減少に伴うより大きな D_v の増加に関連した。I D A 患者は健常者と同様の D_v を有し、大部分の患者が健常者の 1 0 倍を超える、 D_h の劇的な増加を示した。健常者および A C D 患者の正規化臨界体積 v_c は、

30

【 0 1 2 1 】

【数 2 0】

\bar{v}

の約 8 0 %、または約 7 2 f L である。T T または I D A 患者の大半は、通常、

40

【 0 1 2 2 】

【数 2 1】

\bar{v}

および

【 0 1 2 3 】

【数 2 2】

\bar{h}

が減少した。図 2 は、

【 0 1 2 4 】

【数 2 3】

 \bar{v}
および

【0 1 2 5】

【数 2 4】

 \bar{h}

の絶対的な減少に加えて、これらの患者の v_c がさらに減少下し、異なる患者によっては、
るかに大きなばらつきがあったことを示す。

【実施例 3】

10

【0 1 2 6】

鉄欠乏性貧血の予測

本実施例は、平均よりも起点に近い MCHC 線に沿って投影する赤血球の拡大母集団に
基づいて、代償性または潜伏 IDA を、場合によっては、現在可能であるよりも少なくと
も 90 日早く予測することができるという仮説を検査した。投影動作を図 5 に示す。まず
、体積 (v) およびヘモグロビン (h) を有する各赤血球の MCHC 線に沿った投影位置
(u) が判定された。

【0 1 2 7】

【数 2 5】

$$u = v \cdot \cos \theta - h \cdot \sin \theta$$

20

$$\theta = -\tan^{-1}\left(\frac{\bar{h}}{\bar{v}}\right)$$

【0 1 2 8】

MCHC 線に沿った閾値は、平均投影位置の比率 ()

【0 1 2 9】

【数 2 6】

 $\bar{u} : \varphi \cdot \bar{u}$

30

として定義された。次に、起点とこの閾値との間に位置する投影された赤血球分画が以下
のように計算された。式中、 f_{MCHC} は、MCHC 線上の位置の関数としての投影され
た赤血球の確率密度である。

【0 1 3 0】

【数 2 7】

$$P_{\varphi} = \int_0^{\varphi \cdot \bar{u}} f_{MCHC}(u) du$$

40

【0 1 3 1】

図 2 で使用された定常状態 CBC についての異なる閾値の識別効率を比較することによ
り、 $\varphi = 85\%$ の閾値が選択された。図 8 A ~ B に示す 85% の閾値は、他の閾値 (70%
 75% (図 8 C に示す)、 80% 、 90% (図 8 D)、および 100% を含む) と比
較したときに最大の分離を生じさせた。 $P_{0.85}$ の閾値 0.121 が、この訓練集合に
基づいて選択された。

【0 1 3 2】

すべて正常な、新しい独立した患者の 40 の CBC が特定された。これらの正常 CBC
の 20 が、30 ~ 90 日後にさらなる正常 CBC を有した患者からのものであり、これら
の正常 CBC の 20 が、90 日未滿の後に IDA を発症した患者からのものである。2 つ

50

のCBC間に急性出血または鉄補給のある患者は除外された。IDAの定義は、実施例1で前述した。図3A~Dは、 $P_{0.85}$ の閾値0.121が、この独立した被験群の中で75%の感度および100%の特異性を有するIDAを予測することができることを示す。

【実施例4】

【0133】

小球性貧血の鑑別診断

小球性貧血の最も一般的な2つの原因を区別する際の D_h の診断精度を評価するために、図4に示すように、パラメータを10の訓練ケース、すなわちIDAの5ケース、TTの5ケースに適合させた。ケースの定義は、実施例2で前述した。TTの訓練集合は、 $1.7 \times 10^{-15} \sim 2.3 \times 10^{-15}$ の D_h を有した。IDAの訓練集合は、0.009~0.043の D_h を有した。0.0045に等しい閾値、IDA訓練集合の最小 D_h およびTTの訓練集合の最大 D_h の平均が選択された。新しい独立した50のケース、すなわちIDAの22ケース、TTの28ケースがこれにより特定された。22のIDAケースすべてが正確に分類され、28のTTケースのうち27ケースが正確に分類され、総合診断精度は98%であった。この精度は、4つの通常挙げられる判別関数の精度よりも優れている(20)。すなわち、Green&King式の精度は92%、Micro/Hypoは84%、Mentzerは68%、およびEngland&Fraserは57%である。このような他の判別関数のさらなる詳細については参考文献(20)を参照されたい。

(参考文献)

10

20

【化 1】

1. Fauci AS (2008) *Harrison's principles of internal medicine / editors, Anthony S. Fauci ... [et al.]* (McGraw-Hill Medical, New York) 17th Ed pp v. <1-2 >.
2. Donofrio G, *et al.* (1995) Simultaneous Measurement of Reticulocyte and Red-Blood-Cell Indexes in Healthy-Subjects and Patients with Microcytic Anemia. *Blood* 85(3):818-823.
3. Gifford SC, Derganc J, Shevkoplyas SS, Yoshida T, & Bitensky MW (2006) A detailed study of time-dependent changes in human red blood cells: from reticulocyte maturation to erythrocyte senescence. *British Journal of Haematology* 135(3):395-404. 10
4. Waugh RE, *et al.* (1992) Rheologic Properties of Senescent Erythrocytes - Loss of Surface-Area and Volume with Red-Blood-Cell Age. *Blood* 79(5):1351-1358.
5. Willekens FL, *et al.* (2003) Hemoglobin loss from erythrocytes in vivo results from spleen-facilitated vesiculation. *Blood* 101(2):747-751. 20
6. Sens P & Gov N (2007) Force balance and membrane shedding at the red-blood-cell surface. *Phys Rev Lett* 98(1):018102.
7. Willekens FL, *et al.* (2005) Liver Kupffer cells rapidly remove red blood cell-derived vesicles from the circulation by scavenger receptors. *Blood* 105(5):2141-2145.
8. Lew VL, Raftos JE, Sorette M, Bookchin RM, & Mohandas N (1995) Generation of Normal Human Red-Cell Volume, Hemoglobin Content, and Membrane Area Distributions by Birth or Regulation. *Blood* 86(1):334-341. 30

【化 2】

9. Zenker S, Rubin J, & Clermont G (2007) From inverse problems in mathematical physiology to quantitative differential diagnoses. *PLoS Comput. Biol.* 3(11):2072-2086.
10. Zwanzig R (2001) *Nonequilibrium statistical mechanics* (Oxford University Press, Oxford ; New York) pp ix, 222 p.
11. Kampen NGv (1992) *Stochastic processes in physics and chemistry* (North-Holland, Amsterdam ; New York) Rev. and enl. Ed pp xiv, 465 p. 10
12. Lang KS, et al. (2005) Mechanisms of suicidal erythrocyte death. *Cellular Physiology and Biochemistry* 15(5):195-202.
13. Garner C, et al. (2000) Genetic influences on F cells and other hematologic variables: a twin heritability study. *Blood* 95(1):342-346.
14. Robbins SL, Kumar V, & Cotran RS (2010) *Robbins and Cotran pathologic basis of disease* (Saunders/Elsevier, Philadelphia, PA) 8th Ed pp xiv, 1450 p. 20
15. Milbrandt EB, et al. (2006) Predicting late anemia in critical illness. *Crit. Care* 10(1).
16. Rockey DC & Cello JP (1993) Evaluation of the Gastrointestinal-Tract in Patient with Iron-Deficiency Anemia. *N. Engl. J. Med.* 329(23):1691-1695.
17. Lozoff B, Jimenez E, & Wolf AW (1991) Long-Term Developmental Outcome of Infants with Iron-Deficiency. *N. Engl. J. Med.* 325(10):687-694.
18. Zarychanski R & Houston DS (2008) Anemia of chronic disease: A harmful disorder or an adaptive, beneficial response? *Can. Med. Assoc. J.* 179(4):333-337. 30
19. Jopang YP, Thinkhamrop B, Puangpruk R, & Netnee P (2009) False Positive Rates of Thalassemia Screening in Rural Clinical Setting: 10-Year Experience in Thailand. *Southeast Asian J. Trop. Med. Public Health* 40(3):576-580.
20. Ntaios G, et al. (2007) Discrimination indices as screening tests for beta-thalassemic trait. *Ann. Hematol.* 86(7):487-491. 40
21. Lippi G, Salvagno GL, Solero GP, Franchini M, & Guidi GC (2005) Stability of blood cell counts, hematologic parameters and reticulocytes indexes on the Advia A120 hematologic analyzer. *J. Lab. Clin. Med.* 146(6):333-340.

【 0 1 3 4 】

他の実施形態

本発明をその詳細な説明とともに説明したが、前述した説明は例示的なものであり、添付の特許請求の範囲によって定義される本発明の範囲を限定するものではないことを理解

されたい。その他の態様、利点、変更が、以下の特許請求の範囲に含まれる。

【符号の説明】

【0135】

- 700、750 コンピューティング装置
- 702、742、752 プロセッサ
- 704、764 メモリ
- 706 記憶装置
- 708 高速ユーザインターフェース
- 710 高速拡張ポート
- 712 低速ユーザインターフェース
- 714 低速バス
- 720 標準サーバ
- 722 ラップトップコンピュータ
- 724 ラックサーバシステム
- 754 ディスプレイ
- 756 ディスプレイユーザインターフェース
- 758 制御ユーザインターフェース
- 760 音声コーデック
- 762 外部ユーザインターフェース
- 766 通信ユーザインターフェース
- 768 トランシーバ
- 772 拡張ユーザインターフェース
- 774 拡張メモリ
- 780 携帯電話
- 782 スマートフォン

10

20

【図1A】

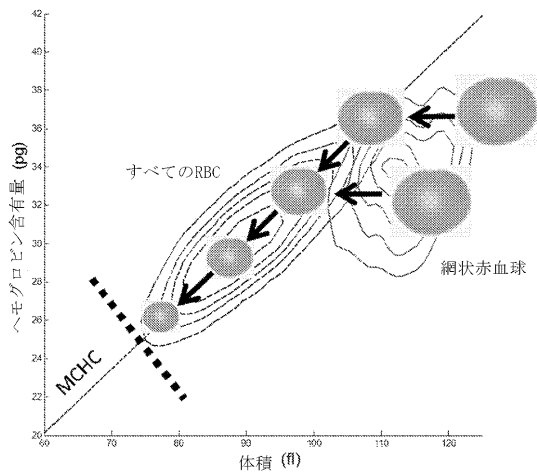


FIG. 1A

【図1B】

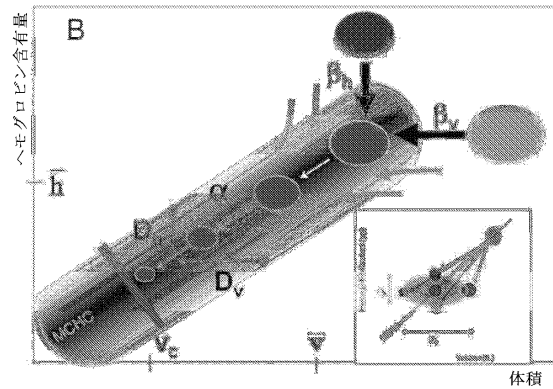


FIG. 1B

【 図 2 】

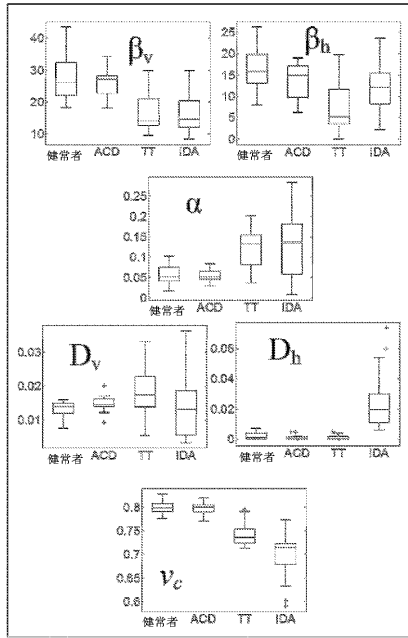


FIG. 2

【 図 3 A 】

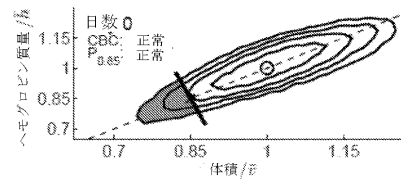


FIG. 3A

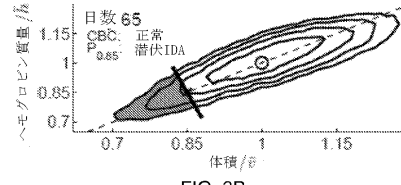


FIG. 3B

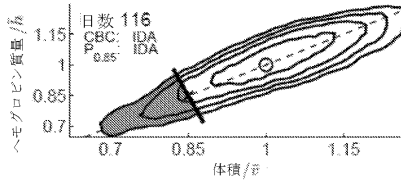


FIG. 3C

【 図 3 B 】

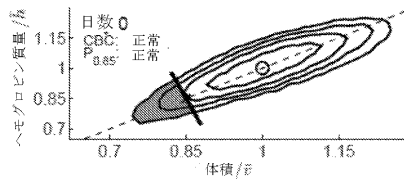


FIG. 3A

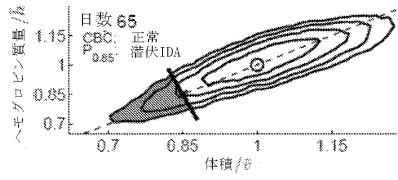


FIG. 3B

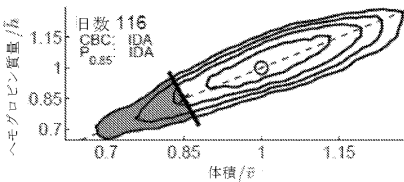


FIG. 3C

【 図 3 C 】

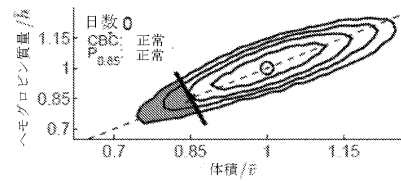


FIG. 3A

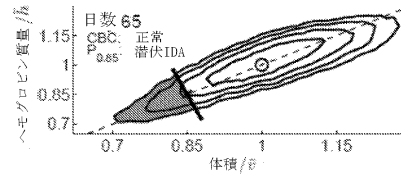


FIG. 3B

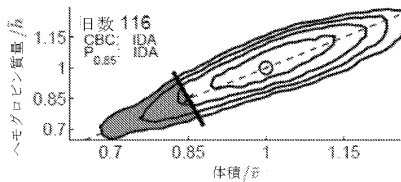


FIG. 3C

【 図 3 D 】

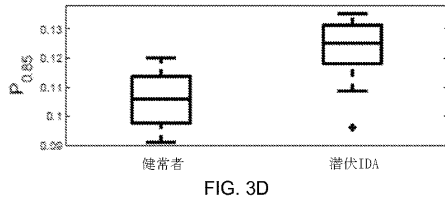


FIG. 3D

【 図 4 】

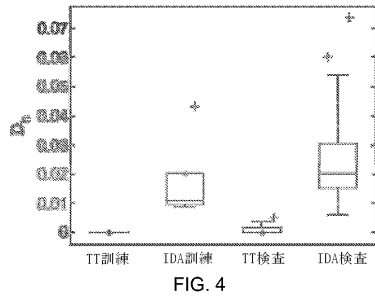


FIG. 4

【 図 5 】

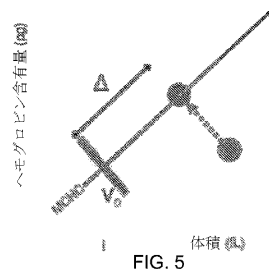


FIG. 5

【 図 6 A 】

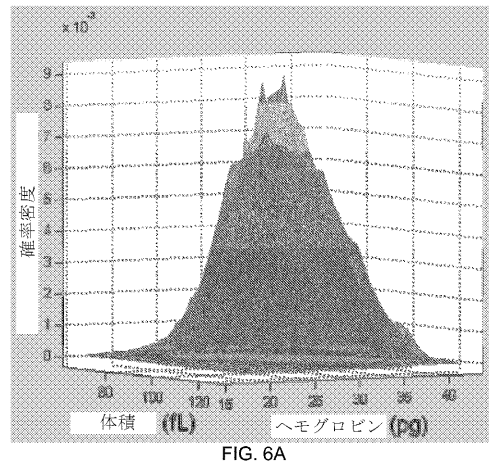


FIG. 6A

【 図 6 B 】

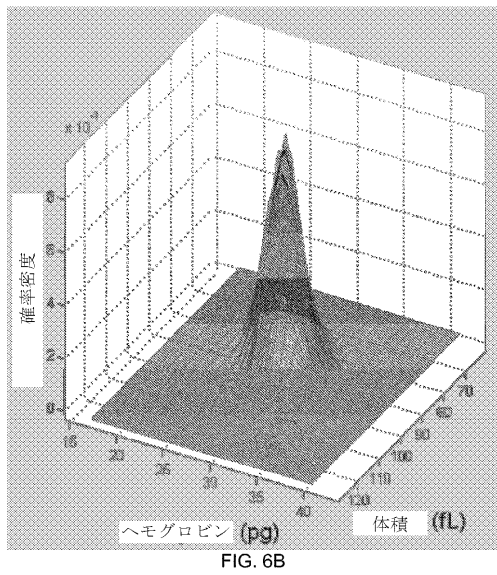
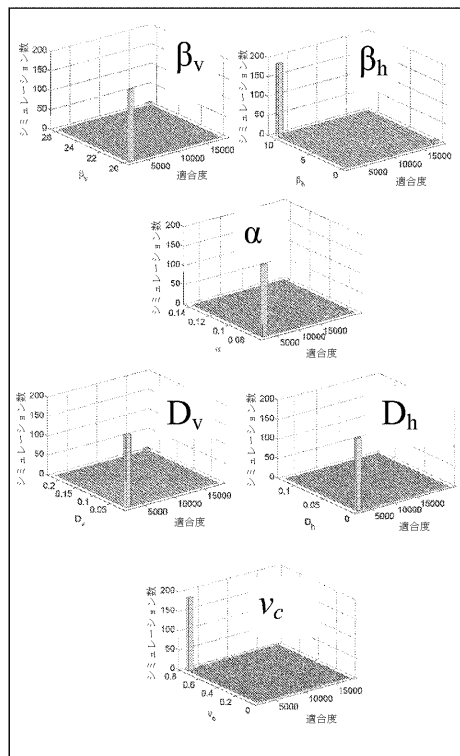


FIG. 6B

【 図 7 】



【 図 8 A 】

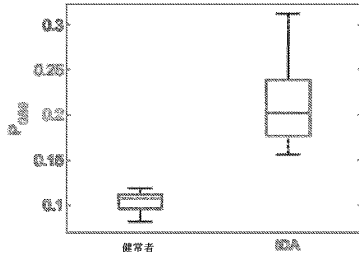


FIG. 8A

【 図 8 B 】

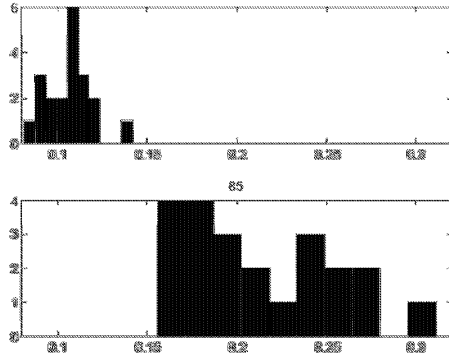


FIG. 8B

【 図 8 C 】

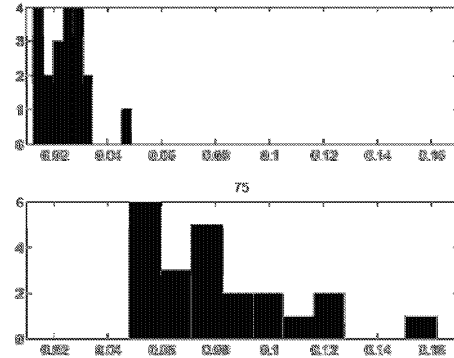


FIG. 8C

【 図 8 D 】

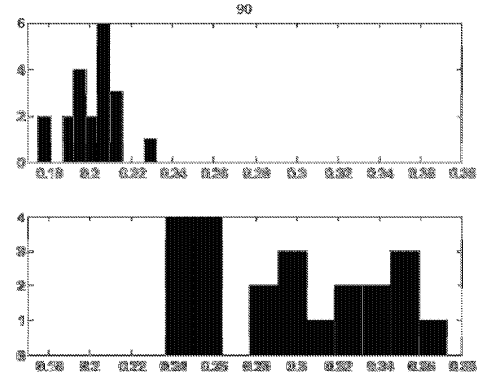
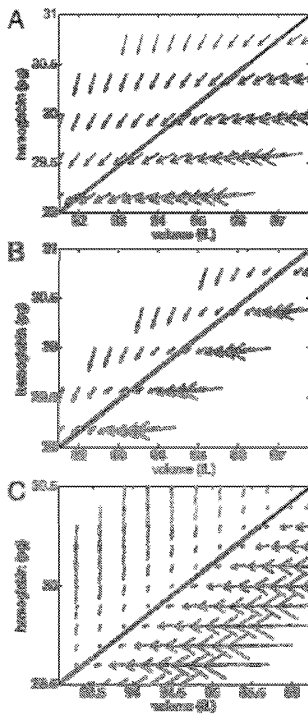


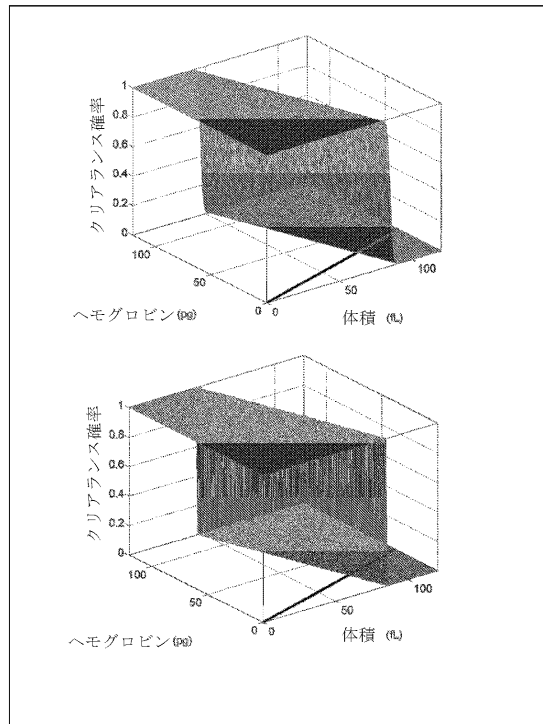
FIG. 8D

【 図 9 】



FIGs. 9A-C

【 図 10 】



FIGs. 10A-B

【図 1 1】

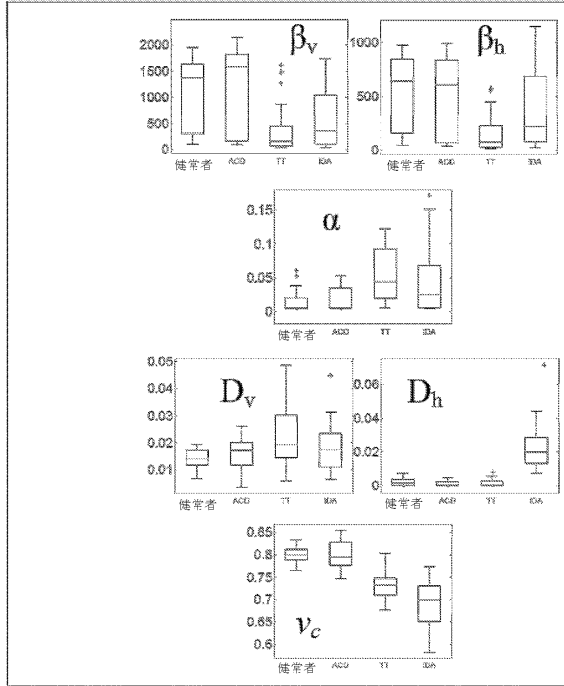


FIG. 11

【図 1 2】

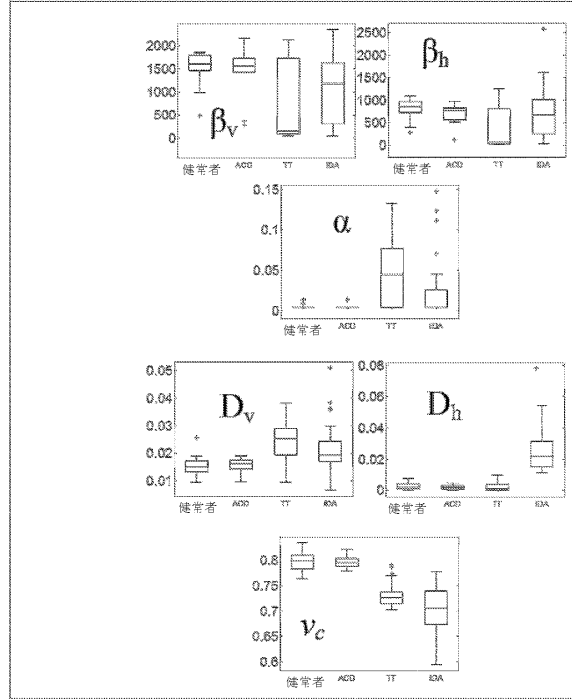


FIG. 12

【図 1 3 A】

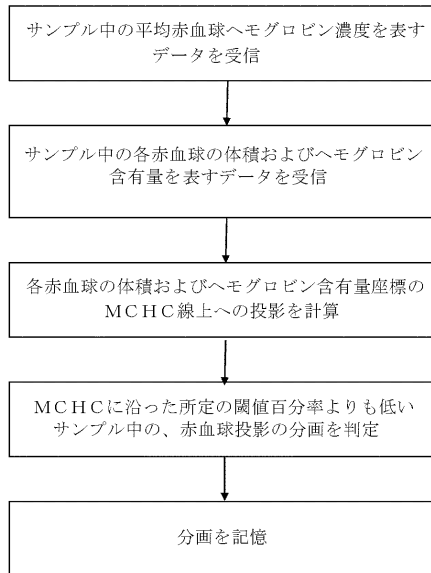


FIG. 13A

【図 1 3 B】

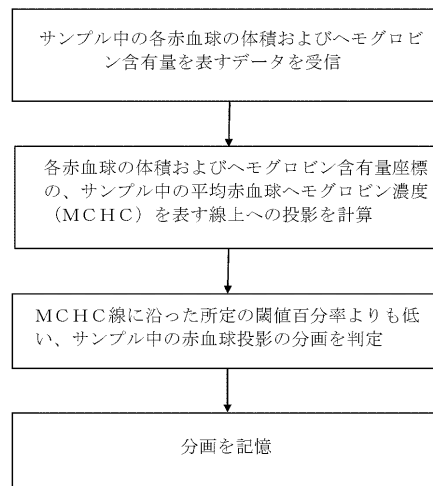


FIG. 13B

【 15 - 4】

```

stAdviaFile.RMCYCFFile = mapFileNames(stAdviaFile.RMCYCFFile);
stAdviaFile.PesticCFFile = mapFileNames(stAdviaFile.PesticCFFile);
stAdviaFile.RawDataFile = mapFileNames(stAdviaFile.RawDataFile);
[MCV, RDW, MCHM, RDW, nData, tSteadyState.vData, tSteadyState.rData] =
getRMCYCF(stAdviaFile, tSteadyState.vData, tSteadyState.rData);
[MCV, RDW, MCHM, RDW, nData, tSteadyState.vData, tSteadyState.rData] =
getRMCYCF(stAdviaFile, tSteadyState.vData, tSteadyState.rData);
tSteadyState.nSteadyStateVersion =
getSteadyStateVersion(stAdviaFile);
tSteadyState.stAdviaFile = stAdviaFile;
end
stSteadyState = tSteadyState;
function sNewName = mapFileNames(sName)
if (~exist('Z:\Research\Microcytois'))
if (~ispc())
if (~isempty(sName))
if (sName) == '\')
sName = ['\Research\Microcytois\' sName];
end
end
sNewName = strrep(sName, 'Z:\Research\Microcytois',
'\Research\Microcytois');
sNewName = strrep(sName, 'C:\Local\Shipping\Research\Microcytois',
'\Research\Microcytois');
sNewName = strrep(sNewName, '\', '/');
else
sNewName = strrep(sName, 'Z:\Research\Microcytois', '.');
if (sName) == '\')
end
sNewName = strrep(sNewName, '\', '/');
end
else
sNewName = sName;
end
end
function stStartingPoints = getStartingPoints(bDB, sDate, stSimulationParameters,
nStartingPoints)
tStartingPoints = [];
if (tStartingPoints == 1)
if (tStartingPoints == sampleLatinHypercube(stSimulationParameters,
nStartingPoints));
else
if (bDB)
stStartingPoints = queryStartingPoints(sDate);
else
stStartingPoints = readStartingPoints(sDate);
end
end
end
function stStartingPoints = readStartingPoints(sDate)
% Read the starting point XLS file to get all possible starting points.
[now, tstr, row] = xlsread('StartingPoint.xls');
stStartingPoints = [];
for ixStart = 1:size(row, 2)
nData = num2cell(stStartingPoints, 1);
stStartingPoint.ID = num2cell(stStartingPoints, 1);
stStartingPoint.nDate = nData;
end
end

```

【 15 - 5】

```

tStartingPoint.alpha = num2cell(stStartingPoints, 1);
tStartingPoint.betaV = num2cell(stStartingPoints, 1);
tStartingPoint.betaH = num2cell(stStartingPoints, 1);
tStartingPoint.Dv = num2cell(stStartingPoints, 1);
tStartingPoint.Dh = num2cell(stStartingPoints, 1);
tStartingPoint.v50 = num2cell(stStartingPoints, 1);
tStartingPoints = [stStartingPoints; tStartingPoints];
end
function stStartingPointsLatin = sampleLatinHypercube(stSimulationParameters,
nStartingPoints)
nDims = 6;
nPoints = nStartingPoints;
stStartingPointsLatin = [];
vStartingPointLatinRow =
strcat(stSimulationParameters().sStartPointMethod, 'Narrow');
bNarrow = all(vStartingPointLatinRow);
% Exponential;
kExponential = strcat(stSimulationParameters().sFunctionalForm,
'Exponential');
bExponential = all(bExponential);
vBijid = strcat(stSimulationParameters().sFunctionalForm, 'Bijid');
bBijid = all(bBijid);
vSimple = strcat(stSimulationParameters().sFunctionalForm, 'Simple');
bSimple = all(bSimple);
vAdditiveExponential = strcat(stSimulationParameters().sFunctionalForm,
'Additive Exponential');
bAdditiveExponential = all(bAdditiveExponential);
vDecayingAdditiveExponential = strcat(stSimulationParameters().sFunctionalForm,
'Decaying Additive Exponential');
bDecayingAdditiveExponential = all(bDecayingAdditiveExponential);
if (bBijid)
if (~bNarrow)
vRangeAlpha = [0 0.1];
vRangeBetaV = [0 50];
vRangeBetaH = [0 50];
vRangeDv = [0 30];
vRangeDh = [0 50];
vRangeV50 = [0 95];
else
vRangeAlpha = [0 0.05];
vRangeBetaV = [0 30];
vRangeBetaH = [0 30];
vRangeDv = [0 10];
vRangeDh = [0 10];
vRangeV50 = [50 95];
end
elseif (bSimple || bExponential || bAdditiveExponential ||
bDecayingAdditiveExponential)
vRangeAlpha = [0 0.05];
vRangeBetaV = [0 500];
vRangeBetaH = [0 500];
vRangeDv = [0 50];
vRangeDh = [0 50];
vRangeV50 = [50 95];
end
end
nIX = lhsdesign(nPoints, nDims);

```

【 15 - 6】

```

for ix = 1:nPoints
stStartingPoint.ID = -1;
tStartingPoint.nDate = datenum(now);
tStartingPoint.alpha = vRangeAlpha(1) + range(vRangeAlpha)*nIX(ix, 1);
tStartingPoint.betaV = vRangeBetaV(1) + range(vRangeBetaV)*nIX(ix, 2);
tStartingPoint.betaH = vRangeBetaH(1) + range(vRangeBetaH)*nIX(ix, 3);
tStartingPoint.Dv = vRangeDv(1) + range(vRangeDv)*nIX(ix, 4);
tStartingPoint.Dh = vRangeDh(1) + range(vRangeDh)*nIX(ix, 5);
tStartingPoint.v50 = vRangeV50(1) + range(vRangeV50)*nIX(ix, 6);
stStartingPointsLatin = [stStartingPointsLatin; tStartingPoints];
end
function stStartingPoints = adjustStartingPoints(stStartingPoints)
vRangeAlpha = range(stStartingPoints.alpha);
vRangeBetaV = range(stStartingPoints.betaV);
vRangeBetaH = range(stStartingPoints.betaH);
vRangeDv = range(stStartingPoints.Dv);
vRangeDh = range(stStartingPoints.Dh);
vRangeV50 = range(stStartingPoints.v50);
for ix = 1:length(stStartingPoints)
stStartingPoint(ix).alpha =
bumpStartingPoint(stStartingPoints(ix).alpha, vRangeAlpha);
stStartingPoint(ix).betaV =
bumpStartingPoint(stStartingPoints(ix).betaV, vRangeBetaV);
stStartingPoint(ix).betaH =
bumpStartingPoint(stStartingPoints(ix).betaH, vRangeBetaH);
stStartingPoint(ix).Dv = bumpStartingPoint(stStartingPoints(ix).Dv,
vRangeDv);
stStartingPoint(ix).Dh = bumpStartingPoint(stStartingPoints(ix).Dh,
vRangeDh);
stStartingPoint(ix).v50 = bumpStartingPoint(stStartingPoints(ix).v50,
vRangeV50);
end
function nStartingPoint = bumpStartingPoint(nStartingPoint, nRange)
nBump = 0.01;
if (nStartingPoint)
nStartingPoint = nStartingPoint + nBump*nRange;
else
nStartingPoint = nStartingPoint + nBump*nRange;
end
end
function stParameters = storeModelParams(bDB, sAccession, stParameters,
stSteadyState, stStartingPoint, stSimulationParameters)
if (~isfield(stParameters.output, 'firstorderopt'))
stParameters.output.firstorderopt = 1;
end
end
nDate = now;
sDate = datestr(nDate, 'mm/dd/yyyy HH:MM:SS PM');
colNamesStart = ['dateEntered', 'alpha', 'beta_v', 'beta_h', 'Dv', 'Dh',
'v50'];
newDataStart = (sDate, stStartingPoint.alpha, stStartingPoint.betaV,
stStartingPoint.betaH, ...
stStartingPoint.Dv, stStartingPoint.Dh,
100*stStartingPoint.v50/stSteadyState.RMW);

```

【 15 - 7】

```

if (bDB)
sODBC = 'microcytois';
z = [];
conn = database(sODBC, '', '');
% Insert the starting point if necessary.
if (stStartingPoint.ID == -1)
fastinsert(conn, 'StartingPoint', colNamesStart, newDataStart);
commit(conn);
curs = exec(conn, ['select max(ID) from StartingPoint']);
curs = fetch(curs);
nStartingPointID = curs.Data;
stStartingPoint.ID = cell2mat(nStartingPointID);
close(curs);
end
end
% Insert the best fit model parameters.
colNames = ['Steady State Parameters', 'Simulation Parameters',
'Starting Point', 'DateEntered', ...
'rho', 'MCV', 'RDW', 'MCH', 'HDM', ...
'muAge', 'alpha', 'alphaH', ...
'betaV', 'betaH', 'v50', 'Dv', 'Dh', ...
'alphaV', 'alphaH', ...
'Dv', 'Dh', ...
'Nu', 'chi2', 'df', 'first order optimality', 'nSlope', 'alphaDecay'];
if (~isfield(stParameters, 'sOptimalTrend'))
stParameters.sOptimalTrend = [];
end
if (strcmp(stParameters.sOptimalTrend, 'Quasi-linear'))
newData = (stSteadyState.ID, stSimulationParameters.ID,
stStartingPoint.ID, datestr(now), ...
stParameters.z, stParameters.mcv, stParameters.rdw, stParameters.mch,
stParameters.hdw, ...
stParameters.muAge, stParameters.x(1), stParameters.x(7), ...
stParameters.x(2), stParameters.x(3), stParameters.x(4),
stParameters.x(5), stParameters.x(6), ...
stParameters.x(1)*stParameters.muAge,
stParameters.x(7)*stParameters.muAge, ...
stParameters.x(5)*stParameters.muAge/stParameters.mcv^2, ...
stParameters.x(4)/stParameters.mcv, stParameters.resnorm,
stParameters.df, stParameters.output.firstorderopt, stParameters.nSlope,
stParameters.alphaDecay);
else
newData = (stSteadyState.ID, stSimulationParameters.ID,
stStartingPoint.ID, datestr(now), ...
stParameters.z, stParameters.mcv, stParameters.rdw, stParameters.mch,
stParameters.hdw, ...
stParameters.muAge, stParameters.x(1), stParameters.x(1), ...
stParameters.x(2), stParameters.x(3), stParameters.x(4),
stParameters.x(5), stParameters.x(6), ...
stParameters.x(1)*stParameters.muAge,
stParameters.x(1)*stParameters.muAge, ...
stParameters.x(5)*stParameters.muAge/stParameters.mcv^2, ...
stParameters.x(6)*stParameters.muAge/stParameters.mch^2, ...
stParameters.x(4)/stParameters.mcv, stParameters.resnorm,
stParameters.df, stParameters.output.firstorderopt, stParameters.nSlope,
stParameters.alphaDecay);
end
end

```

【 15 - 6】

```

for ix = 1:nPoints
stStartingPoint.ID = -1;
tStartingPoint.nDate = datenum(now);
tStartingPoint.alpha = vRangeAlpha(1) + range(vRangeAlpha)*nIX(ix, 1);
tStartingPoint.betaV = vRangeBetaV(1) + range(vRangeBetaV)*nIX(ix, 2);
tStartingPoint.betaH = vRangeBetaH(1) + range(vRangeBetaH)*nIX(ix, 3);
tStartingPoint.Dv = vRangeDv(1) + range(vRangeDv)*nIX(ix, 4);
tStartingPoint.Dh = vRangeDh(1) + range(vRangeDh)*nIX(ix, 5);
tStartingPoint.v50 = vRangeV50(1) + range(vRangeV50)*nIX(ix, 6);
stStartingPointsLatin = [stStartingPointsLatin; tStartingPoints];
end
function stStartingPoints = adjustStartingPoints(stStartingPoints)
vRangeAlpha = range(stStartingPoints.alpha);
vRangeBetaV = range(stStartingPoints.betaV);
vRangeBetaH = range(stStartingPoints.betaH);
vRangeDv = range(stStartingPoints.Dv);
vRangeDh = range(stStartingPoints.Dh);
vRangeV50 = range(stStartingPoints.v50);
for ix = 1:length(stStartingPoints)
stStartingPoint(ix).alpha =
bumpStartingPoint(stStartingPoints(ix).alpha, vRangeAlpha);
stStartingPoint(ix).betaV =
bumpStartingPoint(stStartingPoints(ix).betaV, vRangeBetaV);
stStartingPoint(ix).betaH =
bumpStartingPoint(stStartingPoints(ix).betaH, vRangeBetaH);
stStartingPoint(ix).Dv = bumpStartingPoint(stStartingPoints(ix).Dv,
vRangeDv);
stStartingPoint(ix).Dh = bumpStartingPoint(stStartingPoints(ix).Dh,
vRangeDh);
stStartingPoint(ix).v50 = bumpStartingPoint(stStartingPoints(ix).v50,
vRangeV50);
end
function nStartingPoint = bumpStartingPoint(nStartingPoint, nRange)
nBump = 0.01;
if (nStartingPoint)
nStartingPoint = nStartingPoint + nBump*nRange;
else
nStartingPoint = nStartingPoint + nBump*nRange;
end
end
function stParameters = storeModelParams(bDB, sAccession, stParameters,
stSteadyState, stStartingPoint, stSimulationParameters)
if (~isfield(stParameters.output, 'firstorderopt'))
stParameters.output.firstorderopt = 1;
end
end
nDate = now;
sDate = datestr(nDate, 'mm/dd/yyyy HH:MM:SS PM');
colNamesStart = ['dateEntered', 'alpha', 'beta_v', 'beta_h', 'Dv', 'Dh',
'v50'];
newDataStart = (sDate, stStartingPoint.alpha, stStartingPoint.betaV,
stStartingPoint.betaH, ...
stStartingPoint.Dv, stStartingPoint.Dh,
100*stStartingPoint.v50/stSteadyState.RMW);

```

【 15 - 7】

```

if (bDB)
sODBC = 'microcytois';
z = [];
conn = database(sODBC, '', '');
% Insert the starting point if necessary.
if (stStartingPoint.ID == -1)
fastinsert(conn, 'StartingPoint', colNamesStart, newDataStart);
commit(conn);
curs = exec(conn, ['select max(ID) from StartingPoint']);
curs = fetch(curs);
nStartingPointID = curs.Data;
stStartingPoint.ID = cell2mat(nStartingPointID);
close(curs);
end
end
% Insert the best fit model parameters.
colNames = ['Steady State Parameters', 'Simulation Parameters',
'Starting Point', 'DateEntered', ...
'rho', 'MCV', 'RDW', 'MCH', 'HDM', ...
'muAge', 'alpha', 'alphaH', ...
'betaV', 'betaH', 'v50', 'Dv', 'Dh', ...
'alphaV', 'alphaH', ...
'Dv', 'Dh', ...
'Nu', 'chi2', 'df', 'first order optimality', 'nSlope', 'alphaDecay'];
if (~isfield(stParameters, 'sOptimalTrend'))
stParameters.sOptimalTrend = [];
end
if (strcmp(stParameters.sOptimalTrend, 'Quasi-linear'))
newData = (stSteadyState.ID, stSimulationParameters.ID,
stStartingPoint.ID, datestr(now), ...
stParameters.z, stParameters.mcv, stParameters.rdw, stParameters.mch,
stParameters.hdw, ...
stParameters.muAge, stParameters.x(1), stParameters.x(7), ...
stParameters.x(2), stParameters.x(3), stParameters.x(4),
stParameters.x(5), stParameters.x(6), ...
stParameters.x(1)*stParameters.muAge,
stParameters.x(7)*stParameters.muAge, ...
stParameters.x(5)*stParameters.muAge/stParameters.mcv^2, ...
stParameters.x(4)/stParameters.mcv, stParameters.resnorm,
stParameters.df, stParameters.output.firstorderopt, stParameters.nSlope,
stParameters.alphaDecay);
else
newData = (stSteadyState.ID, stSimulationParameters.ID,
stStartingPoint.ID, datestr(now), ...
stParameters.z, stParameters.mcv, stParameters.rdw, stParameters.mch,
stParameters.hdw, ...
stParameters.muAge, stParameters.x(1), stParameters.x(1), ...
stParameters.x(2), stParameters.x(3), stParameters.x(4),
stParameters.x(5), stParameters.x(6), ...
stParameters.x(1)*stParameters.muAge,
stParameters.x(1)*stParameters.muAge, ...
stParameters.x(5)*stParameters.muAge/stParameters.mcv^2, ...
stParameters.x(6)*stParameters.muAge/stParameters.mch^2, ...
stParameters.x(4)/stParameters.mcv, stParameters.resnorm,
stParameters.df, stParameters.output.firstorderopt, stParameters.nSlope,
stParameters.alphaDecay);
end
end

```

【 15 - 8 】

```

if (bDB)
fastinsert(conn, 'ModelParameters', colNames, newData);
close(conn);
else
newDataStart{1} = datenum(newDataStart{1});
newData{4} = datenum(newData{4});
t = cell2mat(newDataStart, newData);
timestamp = [datestr(now, 'yyyymmdd') ' ' datestr(now, 'HHMMSS')];
dlmwrite([sAccession ' ' timestamp '.txt'], t, '-append', 'precision',
10);
end
end

function stParameters = findOptimalParams(stSteadyState,
stStartingPoint, stSimulationParameters, bDB, sAccession, bParallel)
stParameters = getRBCPDFParameters();

stParameters.nMeshWidth = stSimulationParameters.nMeshWidth;
stParameters.nMinVolume = stSimulationParameters.nMinV;
stParameters.nMaxVolume = stSimulationParameters.nMaxV;
stParameters.nMinHb = stSimulationParameters.nMinH;
stParameters.nMaxHb = stSimulationParameters.nMaxH;
[V, H] =
meshgrid(stParameters.nMinVolume:stParameters.nMeshWidth:stParameters.nMaxVolume
], ...
[stParameters.nMinHb:stParameters.nMeshWidth:stParameters.nMaxHb]);
H = fixpud(H);
stParameters.V = V;
stParameters.H = H;

vEmpirical = strcmp([stSimulationParameters{:}.sInitialDistribution]',
'Empirical');
bEmpirical = all(vEmpirical);
if (bEmpirical)
stParameters.nMeanNarrowVolume = stSteadyState.rMCV;
stParameters.nCVNarrowVolume = stSteadyState.rRDW;
stParameters.nMeanNarrowHb = stSteadyState.rMCH;
stParameters.nCVNarrowHb = stSteadyState.rHDW;
stParameters.vStar = stSteadyState.MCV;
stParameters.hStar = stSteadyState.MCH;
else
stParameters.nMeanNarrowVolume = stSteadyState.stAdviaFile.rMCV;
stParameters.nCVNarrowVolume = stSteadyState.stAdviaFile.rRDW;
stParameters.nMeanNarrowHb = stSteadyState.stAdviaFile.rMCH;
stParameters.nCVNarrowHb = stSteadyState.stAdviaFile.rHDW;
stParameters.vStar = stSteadyState.stAdviaFile.MCV;
stParameters.hStar = stSteadyState.stAdviaFile.MCH;
end

stParameters.P_0 = stSteadyState.P_0;

stParameters.nNarrowRho = stSteadyState.rRho;
stParameters.muAgeLarg = stSteadyState.muAge;

stParameters.alphav = stStartingPoint.alpha;
stParameters.alphah = stStartingPoint.alpha;
stParameters.betav = stStartingPoint.betav;
stParameters.betah = stStartingPoint.betah;
stParameters.v50 = stSteadyState.MCV*stStartingPoint.v50/100;
stParameters.Dv = stStartingPoint.Dv;
stParameters.Dh = stStartingPoint.Dh;

```

【 15 - 9 】

```

stParameters.nSlope = 1;
stParameters.alphaDecay = 6;

stParameters.bParallel = bParallel;
stParameters.sOptimizer = stSimulationParameters.sOptimizer;
stParameters.sFunctionalForm = stSimulationParameters.sFunctionalForm;
stParameters.nMinAlpha = stSimulationParameters.nMinAlpha;
stParameters.nMaxAlpha = stSimulationParameters.nMaxAlpha;
stParameters.sDeathFunction = stSimulationParameters.sDeathFunction;
stParameters.sOptimalTrend = stSimulationParameters.sOptimalTrend;
stParameters.sQuasiLinearFit = getQuasiLinearFit(stParameters,
stSteadyState.vData, stSteadyState.nData);
stParameters.bPenalizeFalsePositives =
stSimulationParameters.bPenalizeFalsePositives;

[k, cesnorm, output, jacobian, stParameters] = optPDFDF(stSteadyState.vData,
stSteadyState.nData, stParameters);
stParameters.v = k;
stParameters.output = output;
stParameters.cesnorm = cesnorm;

% Score them.
stParameters = storeModelParams(bDB, sAccession, stParameters, stSteadyState,
stStartingPoint, stSimulationParameters);
end

% Get the simulation parameter sets entered after an input date.
function stSimulationParameters = getSimulationParameters(bDB, sDate, vSimID)
if (bDB)
stSimulationParameters = querySimulationParameters(sDate);
else
stSimulationParameters = readSimulationParameters(sDate);
end

if (~isempty(vSimID))
vID = [stSimulationParameters{:}.ID];
ix = ismember(vID, vSimID);
stSimulationParameters = stSimulationParameters(ix);
else
%%%%%%
%% Pick the simulation approach.
%%
%%
bMQ = 0;
bFMinSearch = 0;
bPatternSearch = 1;
bBounded = 1;

stStartingPointMethod = 'Narrow';
sFunctionalForm = 'Simple'; % Linear alpha, exponential beta, sigmoid
v50.
sFunctionalForm = 'Hybrid'; % Linear alpha, exponential beta, sigmoid
v50.
sFunctionalForm = 'Exponential'; % Exponential alpha,
exponential beta, and sigmoid v50.
sFunctionalForm = 'Simple'; % Linear alpha, exponential beta, relational
v50.

sInitialDistribution = 'MCV';
sInitialDistribution = 'Empirical';

```

【 15 - 10 】

```

ixOpt = [];
if (bLSQ)
csOptimizers = {stSimulationParameters{:}.sOptimizer}';
ixOpt = [ixOpt; find(strcmp(csOptimizers, 'lsqnonlin'))];
end
if (bFMinSearch)
csOptimizers = {stSimulationParameters{:}.sOptimizer}';
ixOpt = [ixOpt; find(strcmp(csOptimizers, 'fminsearch'))];
end
if (bPatternSearch)
csOptimizers = {stSimulationParameters{:}.sOptimizer}';
ixOpt = [ixOpt; find(strcmp(csOptimizers, 'patternsearch'))];
end

stSimulationParameters = stSimulationParameters(ixOpt);

csMethods = {stSimulationParameters{:}.sStartPointMethod}';
ixSim = strcmp(csMethods, sStartingPointMethod);
stSimulationParameters = stSimulationParameters(ixSim);

csFunctions =
cellstr(strvcat(stSimulationParameters{:}.sFunctionalForm));
ixFunction = strcmp(csFunctions, sFunctionalForm);
stSimulationParameters = stSimulationParameters(ixFunction);

csDist = {stSimulationParameters{:}.sInitialDistribution}';
ixDist = strcmp(csDist, sInitialDistribution);
stSimulationParameters = stSimulationParameters(ixDist);

ixNoBounds = isnan([stSimulationParameters{:}.nMinAlpha]);
if (bBounded)
ixBounds = ~ixNoBounds;
stSimulationParameters = stSimulationParameters(ixBounds);
else
stSimulationParameters = stSimulationParameters(ixNoBounds);
end
end
stSimulationParameters

end

function stSimulationParameters = readSimulationParameters(sDate)
% Read the sample XLS file to get the Advia File ID.
[num, txt, raw] =
xlsread('simulation.xls', 'SimulationParameters', '', 'basic');
nOffset = size(txt, 1) - size(num, 1);
stSimulationParameters = [];
for ixRow = 1:size(num,1)
% Read the advia file XLS file to get the FCS location and the steady
% state measurements.
tSimulationParameters.ID = num(ixRow, 1);
tSimulationParameters.sDate = txt{ixRow + nOffset, 2};
tSimulationParameters.sOptimizer = txt{ixRow + nOffset, 3};
tSimulationParameters.nMeshWidth = num{ixRow, 5};
tSimulationParameters.nMinV = num{ixRow, 6};
tSimulationParameters.nMaxV = num{ixRow, 7};
tSimulationParameters.nMinH = num{ixRow, 8};
tSimulationParameters.nMaxH = num{ixRow, 9};
tSimulationParameters.sStartPointMethod = txt{ixRow + nOffset, 10};
tSimulationParameters.sFunctionalForm = txt{ixRow + nOffset, 11};
tSimulationParameters.sInitialDistribution = txt{ixRow + nOffset, 12};
tSimulationParameters.nMinAlpha = num{ixRow, 13};

```

【 15 - 11 】

```

stSimulationParameters.nMaxAlpha = num{ixRow, 14};
stSimulationParameters.sDeathFunction = txt{ixRow + nOffset, 15};
stSimulationParameters.sOptimalTrend = txt{ixRow + nOffset, 16};
stSimulationParameters.nSteadyStateVersion = num{ixRow, 17};
if (~isfinite(stSimulationParameters.nSteadyStateVersion))
stSimulationParameters.nSteadyStateVersion = 0;
end
stSimulationParameters.bPenalizeFalsePositives = num{ixRow, 18};
if (~isfinite(stSimulationParameters.bPenalizeFalsePositives))
stSimulationParameters.bPenalizeFalsePositives = 0;
end
stSimulationParameters = [stSimulationParameters; tSimulationParameters];
end

function stSimulationParameters = querySimulationParameters(sDate)
sQMC = 'Microsoft';
f = [];
conn = database(sQMC, '', '');
curs = exec(conn, ['select * from SimulationParameters']);
curs = fetch(curs);
csSimulationParameters = curs.Data;

stSimulationParameters = [];
for ixStart = 1:size(csSimulationParameters, 1)
nDate = datenum(csSimulationParameters{ixStart, 2});
if (nDate > datenum(sDate))
[SimulationParameters.ID = csSimulationParameters{ixStart, 1};
SimulationParameters.sDate = nDate;
SimulationParameters.sOptimizer = csSimulationParameters{ixStart,
3};
SimulationParameters.nMeshWidth = csSimulationParameters{ixStart,
5};
SimulationParameters.nMinV = csSimulationParameters{ixStart, 6};
SimulationParameters.nMaxV = csSimulationParameters{ixStart, 7};
SimulationParameters.nMinH = csSimulationParameters{ixStart, 8};
SimulationParameters.nMaxH = csSimulationParameters{ixStart, 9};
SimulationParameters.sStartPointMethod =
csSimulationParameters{ixStart, 10};
SimulationParameters.sFunctionalForm =
csSimulationParameters{ixStart, 11};
SimulationParameters.sInitialDistribution =
csSimulationParameters{ixStart, 12};
SimulationParameters.nMinAlpha = csSimulationParameters{ixStart,
13};
SimulationParameters.nMaxAlpha = csSimulationParameters{ixStart,
14};
SimulationParameters.sDeathFunction =
csSimulationParameters{ixStart, 15};
SimulationParameters.sOptimalTrend = csSimulationParameters{ixStart,
16};
SimulationParameters.nSteadyStateVersion =
csSimulationParameters{ixStart, 17};
if (isnan(SimulationParameters.nSteadyStateVersion))
SimulationParameters.nSteadyStateVersion = 0;
end
SimulationParameters.bPenalizeFalsePositives =
csSimulationParameters{ixStart, 18};
if (isnan(SimulationParameters.bPenalizeFalsePositives))
SimulationParameters.bPenalizeFalsePositives = 0;
end
end
end

```

【 15 - 12 】

```

end
stSimulationParameters = [stSimulationParameters;
tSimulationParameters];
end
close(curs);
close(conn);
end
% Get the starting points entered after the threshold date.
function stStartingPoints = queryStartingPoints(sDate)
sDate = 'microcytosis';
k = [];
conn = database(sODBC, '', '');
curs = exec(conn, ['select * from StartingPoint']);
curs = fetch(curs);
csStartingPoint = curs.Data;
stStartingPoints = [];
for ixStart = 1:size(csStartingPoint, 2)
    nDate = getNumericStartingPoint(ixStart, 2);
    if (nDate > determineState)
        tStartingPoint.ID = csStartingPoint(ixStart, 1);
        tStartingPoint.nDate = nDate;
        tStartingPoint.Alpha = csStartingPoint(ixStart, 3);
        tStartingPoint.beta = csStartingPoint(ixStart, 4);
        tStartingPoint.betah = csStartingPoint(ixStart, 5);
        tStartingPoint.Dw = csStartingPoint(ixStart, 6);
        tStartingPoint.Dh = csStartingPoint(ixStart, 7);
        tStartingPoint.v50 = csStartingPoint(ixStart, 8);
        stStartingPoints = [stStartingPoints; tStartingPoint];
    end
end
close(curs);
close(conn);
end
function [Phat, P, vi, hi] = makeEmpiricalPDF(vData, hcData)
P = [];
% Convert hemoglobin content (mass), not concentration.
hData = vData.*hcData/100;
% Assume the true RBC population distribution is bivariate normal.
% Approximate the true distribution from a sampling of the sample.
vi = linspace(min(vData), max(vData), round(range(vData)));
hi = linspace(min(hData), max(hData), round(range(hData)));
ctrx(1) = vi;
ctrx(2) = hi;
% Calculate the empirical distribution.
P = hist3(vData, hData, ctrx);
P = flipud(P);
hi = flipud(hi);
vi = repmat(vi(1), size(P, 1), 1);
hi = repmat(hi(1), 1, size(P, 2));
% Also approximate the true distribution by smoothing the raw data.
n = 3;
nCenterWeight = 1/n;
k = (1 - nCenterWeight)^(1/(n^2 - 1))*ones(n,n);

```

【 15 - 13 】

```

k(ceil(n/2),ceil(n/2)) = nCenterWeight;
Phat = conv2(P, k, 'same');
Phat = sum(Phat(:))*Phat/sum(Phat(:));
ixCells = find(P > 5);
%Phat = sum(Phat(ixCells))*Phat/sum(Phat(ixCells));
%Chi2 = sum(P(ixCells) - Phat(ixCells)).^2./P(ixCells);
end
function xi = getNormSpace(x)
mu = endata(s);
sigma = sqrt(x)*0.7314;
normx = linspace((max(x) - mu)/sigma, (max(x) + mu)/sigma, round(range(x)));
xi = abs(2)*normx.*exp(-1*normx.^2);
xi = xi/sum(xi);
xi = cumsum(xi);
xi = [0 xi];
xi = xi*range(x) + min(x);
end
function stSteadyState = queryFCSSteadyState(sAccession, sDate, nVersion)
% Given an accession number, return, calculate, and populate the steady
% state parameters after a given date as necessary.
sODBC = 'microcytosis';
r = [];
conn = database(sODBC, '', '');
% Get all sample IDs.
nSampleID = getSampleID(conn, sAccession);
if (~isempty(nSampleID))
% Get the ADVIA file names and measured results.
stAdviaFile = getAdviaFile(conn, nSampleID);
% Get the calculated steady state values.
stSteadyState = getCalculatedSteadyState(conn, stAdviaFile);
% Get those calculated after a certain date.
stSteadyState = getCurrentSteadyState(stSteadyState, sDate, nVersion);
% If there's no fresh steady state:
if (isempty(stSteadyState))
% Calculate and store the steady state if not already stored.
stSteadyState = calculateSteadyState(conn, stAdviaFile, nVersion);
stSteadyState = getCalculatedSteadyState(conn, stAdviaFile);
stSteadyState = getCurrentSteadyState(stSteadyState, sDate,
nVersion);
end
% Add the advia-calculated parameters in case the simulation wants
% to use them.
for ix = 1:numel(stSteadyState)
    stSteadyState(ix).stAdviaFile = stAdviaFile;
end
end
close(conn);
end

```

【 15 - 14 】

```

function stSteadyState = calculateSteadyState(conn, stAdviaFile, nVersion)
stSteadyState = [];
% Calculate the moments of the RBC distribution.
[nMCV, nrRDW, nrCHCM, nrHb, nrHw, v, h, nVersion] = getRBCPDF(stAdviaFile,
nVersion);
% Calculate the moments of the reticulocyte distribution.
[f, nrMCV, nrRDW, nrCHCM, nrHb, nrHw, vr, hr, nVersion] =
getReticPDF(stAdviaFile, nVersion);
nDate = now;
sDate = datestr(nDate, 'mm/dd/yyyy HH:MM:SS PM');
collNames = ['DateEntered', 'AdviaFile', 'MCV', 'RDW', 'MCH', 'RDW', 'rhc',
...
'fraction', 'nMCV', 'nrRDW', 'nrCHCM', 'nrHb', 'nrHw', 'nVersion'];
newData = [sDate, stAdviaFile.ID, nMCV, nrRDW/100, nrCHCM, nrHb/100, nrHw(1,2),
...
f, nrMCV, nrRDW/100, nrCHCM, nrHb/100, nrHw(1,2), nVersion];
insert(conn, 'SteadyStateParameters', collNames, newData);
commit(conn);
curs = exec(conn, ['select max(ID) from SteadyStateParameters']);
curs = fetch(curs);
nID = curs.Data;
stSteadyState.ID = cell2mat(nID);
end
% Return a list of steady state calculations with dates after an input and
% sorted.
function stSteadyState = getCurrentSteadyState(stSteadyState, sDate, nVersion)
if (isfinite(nVersion))
if (numel(stSteadyState))
vVersions = [stSteadyState(:,nSteadyStateVersion);
ix = find(vVersions == nVersion);
stSteadyState = stSteadyState(ix);
end
end
if numel(stSteadyState)
vDates = [stSteadyState(:,nDate)];
ixCurrent = find(vDates > determineState);
[iv, ix] = sort([stSteadyState(ixCurrent,nDate)];
stSteadyState = stSteadyState(ixCurrent:ix);
stSteadyState = stSteadyState(end);
end
end
function [f, nrMCV, nrRDW, nrCHCM, nrHb, nrHw, vr, hcr, nVersion] = getRBCPDF(stAdviaFile,
nVersion)
if ( nargin < 2)
nVersion = 0;
end
if (isstruct(stAdviaFile))
sFCS = stAdviaFile.ReticFCSFile;
else
sFCS = stAdviaFile;
end

```

【 15 - 15 】

```

if (~isempty(sFCS) && ~strcmp(sFCS, 'null'))
% Old version using FCS files.
[nMCV, nrRDW, nrCHCM, nrHb, nrHw, nrHw, v, hc, nVersion] =
compareMieTransforms(sFCS, nVersion);
else
% Assume it's an Abbott-supplied file.
[nMCV, nrRDW, nrCHCM, nrHb, nrHw, v, hc] = getCSVPDF(stAdviaFile);
end
end
function [nMCV, nrRDW, nrCHCM, nrHb, nrHw, v, hc] = getCSVPDF(stAdviaFile)
[n, t, r] = xlsread(stAdviaFile.RawDataFile);
ixRetic = (n(:,1) == 1);
ixRBC = (n(:,1) == 2);
v = n(ixRBC, 3);
hc = n(ixRBC, 2);
h = v.*hc/100;
nMCH = mean(h);
nMCV = mean(v);
nrRDW = 100*std(v)/nMCV;
nrHb = 100*std(h)/nMCH;
nrHw = corrcorr([v h]);
end
function [f, nrMCV, nrRDW, nrCHCM, nrHb, nrHw, vr, hcr, nVersion] =
getReticPDF(stAdviaFile, nVersion)
if ( nargin < 2)
nVersion = 0;
end
if (isstruct(stAdviaFile))
sFCS = stAdviaFile.ReticFCSFile;
else
sFCS = stAdviaFile;
end
if (~isempty(sFCS) && ~strcmp(sFCS, 'null'))
% Old version using FCS files.
[f, nrMCV, nrRDW, nrCHCM, nrHb, nrHw, vr, hcr, nVersion] =
estimateReticFraction(sFCS, nVersion);
else
% Assume it's an Abbott-supplied file.
[f, nrMCV, nrRDW, nrCHCM, nrHb, nrHw, vr, hcr] =
getCSVReticPDF(stAdviaFile);
end
end
function [f, nrMCV, nrRDW, nrCHCM, nrHb, nrHw, vr, hcr] =
getCSVReticPDF(stAdviaFile)
[n, t, r] = xlsread(stAdviaFile.RawDataFile);
ixRetic = (n(:,1) == 1);
ixRBC = (n(:,1) == 2);
vr = n(ixRetic, 3);
hcr = n(ixRetic, 2);
hr = vr.*hcr/100;
nrCHCM = mean(hr);
nrMCV = mean(vr);

```


【 15 - 16 】

```

nrRDW = 100*std(vz)/nrMCV;
nrHDW = 100*std(hz)/nrCHC;
nrRho = corrcoeff(vr hr);
f = 100*sum(ixRetic)/sum(ixRBC);
end

function [f, nMCV, nrRDW, nrCHC, nrHDW, nrRho, v, h, nVersion] =
estimateReticFraction(sFCS, nVersion, nSigmaMin, nSigmaMax)
if (nargin < 3)
    nSigmaMin = (2*sqrt(2*log(2)));
    nSigmaMax = Inf;
    if (nargin < 2)
        nVersion = 0;
    end
end

if (~exist(sFCS))
    sNewName = strrep(sFCS, 'Z:\Research\Microcytosis',
'C:\local\john\BWH\Microcytosis');
    if (~exist(sNewName))
        sNewName = strrep(sFCS, 'Z:\Research\Microcytosis',
'C:\local\jhhiggins\research\microcytosis');
    end
    sFCS = sNewName;
end

[d,p,h] = fcsread(sFCS);
m = makeMieLookupTable;

if (nVersion == 0)
    d = filterRaw(d);
    [v, h, ix] = transformMie(m, d, 5);
    d = d(ix, :);
    [v, h, ix] = filterTransformed(v, h);
    d = d(ix, :);

    n = hist(d(:,3), [1:256]);
    [pMax, ixMode] = max(n);
    ixHigh = find(n >= pMax/2);
    sigma_hat = range(ixHigh)/(2*sqrt(2*log(2)));
    ixV = find(d(:, 3) >= (median(d(:, 3)) + nSigmaMin*sigma_hat));
    if (isfinite(nSigmaMax))
        ixV = intersect(ixV, find(d(:, 3) <= (median(d(:, 3)) +
nSigmaMax*sigma_hat)));
    end
    f = 100*length(ixV)/size(d, 1);

    [v, h, ix] = transformMie(m, d(ixV,:), 5);
    nMCV = mean(v);
    nrRDW = 100*std(v)/mean(v);
    nrCHC = mean(v.*h/100);
    nrHDW = 100*std(100*v.*h)/mean(100*v.*h);
    nrRho = corrcoeff([v v.*h]);
elseif (nVersion == 1)
    stPBANRES = getReticPBANRES(sFCS);
    [v, h, ix, vFull, hFull] = transformMie(m, d, inf);
    n99 = 250 - 2.5;
    ixBZ = find(d(:, 3) >= n99);
    ixNull = (union(find(isnan(hFull)), find(isnan(vFull))));

```

【 15 - 18 】

```

ixRetic = find(d(ixAnalyzed(ixGated), 3) >= nThreshold);
v = vFull(ixAnalyzed(ixGated(ixRetic)));
h = hFull(ixAnalyzed(ixGated(ixRetic)));
f = 100*length(ixRetic)/length(ixGated);
nMCV = mean(v);
nrRDW = 100*std(v)/0.7413/nMCV;
nrCHC = mean(v.*h/100);
nrHDW = 100*std(100*v.*h)/0.7413/mean(100*v.*h);
nrRho = corrcoeff([v v.*h]);
elseif (nVersion == 3)
    stPBANRES = getReticPBANRES(sFCS);
    [v, h, ix, vFull, hFull] = transformMie(m, d, inf);
    n99 = 250 - 2.5;
    ixBZ = find(d(:, 3) >= n99);
    ixNull = (union(find(isnan(hFull)), find(isnan(vFull))));
    ixNotAnalyzed = union(ixNull, ixBZ);
    ixAnalyzed = setdiff(1:size(d, 1), ixNotAnalyzed);

    nBins = 99;
    [xCounts, xBins] = hist(d(:,2), nBins);
    ixX = intersect(find(d(ixAnalyzed,2) >=
mean(xBins(max(stPBANRES.nPLTThreshold, 2) + [-1 0])), find(d(ixAnalyzed,2) <=
mean(xBins(stPBANRES.nMaxX + [0 1]))));

    [yCounts, yBins] = hist(d(:,1), nBins);
    ixY = intersect(find(d(ixAnalyzed,1) > 0), find(d(ixAnalyzed,1) <=
mean(yBins(stPBANRES.nMaxY + [0 1]))));
    ixGated = intersect(ixX, ixY);

    [zCounts, zBins] = hist(d(:,3), nBins);
    ixRetic = (find(d(ixAnalyzed(ixGated), 3) >=
mean(zBins(stPBANRES.nMinRTThreshold + [-1 0]))));
    ixLo = (find(d(ixAnalyzed(ixGated), 3) >=
zBins(stPBANRES.nLoThreshold)));
    ixMed = (find(d(ixAnalyzed(ixGated), 3) >=
zBins(stPBANRES.nMedThreshold)));

    nThreshold = prtile(d(ixAnalyzed(ixGated), 3), 100*(1 -
stPBANRES.nReticCount/stPBANRES.nGatedCount));
    ixRetic = find(d(ixAnalyzed(ixGated), 3) >= nThreshold);

    v = vFull(ixAnalyzed(ixGated(ixRetic)));
    h = hFull(ixAnalyzed(ixGated(ixRetic)));

    h_pg = v.*h/100;
    nMCV = mean(v);
    nrCHC = mean(h_pg);
    nsdV = std(v);
    nsdH_pg = std(h_pg);

    % Shift and scale to match the mean and variance.
    v = (v - nMCV)/nsdV;
    v = (v*stPBANRES.nRDW*stPBANRES.nMCVr/100) + stPBANRES.nMCVr;
    h_pg = (h_pg - nrCHC)/nsdH_pg;
    h_pg = (h_pg*stPBANRES.nCHDWr) + stPBANRES.nCHCr;
    h = 100*h_pg/v;
    nMCV = mean(v);
    nrCHC = mean(v.*h/100);

    nrRDW = 100*std(v)/0.7413/nMCV;
    nrHDW = 100*std(100*v.*h)/0.7413/mean(100*v.*h);
    nrRho = corrcoeff([v v.*h]);

    f = 100*length(ixRetic)/length(ixGated);
end
end

function oQuasiLinearFit = getQuasiLinearFit(stParameters, vData, hcData)
qft = fitype('a*x+b');
oQuasiLinearFit = fit(vData, vData.*hcData/100, qft);
end

function [stParameters, nSampleID] = getRBCPDFParameters(nModelParametersID)

```

【 15 - 17 】

```

ixNotAnalyzed = union(ixNull, ixBZ);
ixAnalyzed = setdiff(1:size(d, 1), ixNotAnalyzed);

nBins = 99;
[xCounts, xBins] = hist(d(:,2), nBins);
ixX = intersect(find(d(ixAnalyzed,2) >=
mean(xBins(max(stPBANRES.nPLTThreshold, 2) + [-1 0])), find(d(ixAnalyzed,2) <=
mean(xBins(stPBANRES.nMaxX + [0 1]))));

[yCounts, yBins] = hist(d(:,1), nBins);
ixY = intersect(find(d(ixAnalyzed,1) > 0), find(d(ixAnalyzed,1) <=
mean(yBins(stPBANRES.nMaxY + [0 1]))));
ixGated = intersect(ixX, ixY);

[xCounts, xBins] = hist(d(:,3), nBins);
ixRetic = (find(d(ixAnalyzed(ixGated), 3) >=
mean(xBins(stPBANRES.nMinRTThreshold + [-1 0]))));
ixLo = (find(d(ixAnalyzed(ixGated), 3) >=
xBins(stPBANRES.nLoThreshold)));
ixMed = (find(d(ixAnalyzed(ixGated), 3) >=
xBins(stPBANRES.nMedThreshold)));

v = vFull(ixAnalyzed(ixGated(ixRetic)));
h = hFull(ixAnalyzed(ixGated(ixRetic)));
f = 100*length(ixRetic)/length(ixGated);
nMCV = mean(v);
nrRDW = 100*std(v)/0.7413/nMCV;
nrCHC = mean(v.*h/100);
nrHDW = 100*std(100*v.*h)/0.7413/mean(100*v.*h);
nrRho = corrcoeff([v v.*h]);
elseif (nVersion == 2)
    stPBANRES = getReticPBANRES(sFCS);
    [v, h, ix, vFull, hFull] = transformMie(m, d, inf);
    n99 = 250 - 2.5;
    ixBZ = find(d(:, 3) >= n99);
    ixNull = (union(find(isnan(hFull)), find(isnan(vFull))));
    ixNotAnalyzed = union(ixNull, ixBZ);
    ixAnalyzed = setdiff(1:size(d, 1), ixNotAnalyzed);

    nBins = 99;
    [xCounts, xBins] = hist(d(:,2), nBins);
    ixX = intersect(find(d(ixAnalyzed,2) >=
mean(xBins(max(stPBANRES.nPLTThreshold, 2) + [-1 0])), find(d(ixAnalyzed,2) <=
mean(xBins(stPBANRES.nMaxX + [0 1]))));

    [yCounts, yBins] = hist(d(:,1), nBins);
    ixY = intersect(find(d(ixAnalyzed,1) > 0), find(d(ixAnalyzed,1) <=
mean(yBins(stPBANRES.nMaxY + [0 1]))));
    ixGated = intersect(ixX, ixY);

    [zCounts, zBins] = hist(d(:,3), nBins);
    ixRetic = (find(d(ixAnalyzed(ixGated), 3) >=
mean(xBins(stPBANRES.nMinRTThreshold + [-1 0]))));
    ixLo = (find(d(ixAnalyzed(ixGated), 3) >=
zBins(stPBANRES.nLoThreshold)));
    ixMed = (find(d(ixAnalyzed(ixGated), 3) >=
zBins(stPBANRES.nMedThreshold)));

    nThreshold = prtile(d(ixAnalyzed(ixGated), 3), 100*(1 -
stPBANRES.nReticCount/stPBANRES.nGatedCount));

```

【 15 - 19 】

```

ixNotAnalyzed = union(ixNull, ixBZ);
ixAnalyzed = setdiff(1:size(d, 1), ixNotAnalyzed);

nBins = 99;
[xCounts, xBins] = hist(d(:,2), nBins);
ixX = intersect(find(d(ixAnalyzed,2) >=
mean(xBins(max(stPBANRES.nPLTThreshold, 2) + [-1 0])), find(d(ixAnalyzed,2) <=
mean(xBins(stPBANRES.nMaxX + [0 1]))));

[yCounts, yBins] = hist(d(:,1), nBins);
ixY = intersect(find(d(ixAnalyzed,1) > 0), find(d(ixAnalyzed,1) <=
mean(yBins(stPBANRES.nMaxY + [0 1]))));
ixGated = intersect(ixX, ixY);

[zCounts, zBins] = hist(d(:,3), nBins);
ixRetic = (find(d(ixAnalyzed(ixGated), 3) >=
mean(zBins(stPBANRES.nMinRTThreshold + [-1 0]))));
ixLo = (find(d(ixAnalyzed(ixGated), 3) >=
zBins(stPBANRES.nLoThreshold)));
ixMed = (find(d(ixAnalyzed(ixGated), 3) >=
zBins(stPBANRES.nMedThreshold)));

nThreshold = prtile(d(ixAnalyzed(ixGated), 3), 100*(1 -
stPBANRES.nReticCount/stPBANRES.nGatedCount));
ixRetic = find(d(ixAnalyzed(ixGated), 3) >= nThreshold);

v = vFull(ixAnalyzed(ixGated(ixRetic)));
h = hFull(ixAnalyzed(ixGated(ixRetic)));

h_pg = v.*h/100;
nMCV = mean(v);
nrCHC = mean(h_pg);
nsdV = std(v);
nsdH_pg = std(h_pg);

% Shift and scale to match the mean and variance.
v = (v - nMCV)/nsdV;
v = (v*stPBANRES.nRDW*stPBANRES.nMCVr/100) + stPBANRES.nMCVr;
h_pg = (h_pg - nrCHC)/nsdH_pg;
h_pg = (h_pg*stPBANRES.nCHDWr) + stPBANRES.nCHCr;
h = 100*h_pg/v;
nMCV = mean(v);
nrCHC = mean(v.*h/100);

nrRDW = 100*std(v)/0.7413/nMCV;
nrHDW = 100*std(100*v.*h)/0.7413/mean(100*v.*h);
nrRho = corrcoeff([v v.*h]);

f = 100*length(ixRetic)/length(ixGated);
end
end

function oQuasiLinearFit = getQuasiLinearFit(stParameters, vData, hcData)
qft = fitype('a*x+b');
oQuasiLinearFit = fit(vData, vData.*hcData/100, qft);
end

function [stParameters, nSampleID] = getRBCPDFParameters(nModelParametersID)

```

【 15 - 20 】

```

nSampleID = 1;
if ( nargin == 0 )
    stParameters.nMeshWidth = 1.8;
    stParameters.nMaxVolume = 180;
    stParameters.nMinVolume = 30;
    stParameters.nMaxHb = 60;
    stParameters.nMinHb = 0;
    [V,H] = meshgrid(stParameters.nMinVolume:stParameters.nMeshWidth:stParameters.nMaxVolume
); ...
    [stParameters.nMinHb:stParameters.nMeshWidth:stParameters.nMaxHb];
    H = flipud(H);
    stParameters.V = V;
    stParameters.H = H;

    stParameters.sOptimizer = 'lsqnonlin';
    stParameters.sDeathFunction = 'Dimensionless Projection';
    stParameters.sFunctionalForm = 'Additive Exponential';
    stParameters.bParallel = 0;
    stParameters.nMatchParameters = 0;
    stParameters.bPenalizeFalsePositives = 0;
    stParameters.nMinAlpha = NaN;
    stParameters.nMaxAlpha = NaN;

    stParameters.nMeanMarrowVolume = 109.3;
    stParameters.nCVMarrowVolume = 0.1129;
    stParameters.nMeanMarrowHb = 32.57;
    stParameters.nCVMarrowHb = 0.1011;
    stParameters.nMarrowRho = 0.34;
    stParameters.nDays = 0;
    stParameters.nHStar = 30.04;
    stParameters.nDays = 0;
    stParameters.nMuAgeTarg = 50;

    sDF = (stParameters.nCVMarrowVolume*stParameters.nMeanMarrowVolume);
    sDR = (stParameters.nCVMarrowHb*stParameters.nMeanMarrowHb);
    nCovar = (stParameters.nMarrowRho*sDF*sDR);
    nSigma = [sDF^2 nCovar; nCovar sDR^2];
    P_0 = mvnpdf([V(:) H(:)], [stParameters.nMeanMarrowVolume
    stParameters.nMeanMarrowHb], nSigma);
    P_0 = P_0./sum(P_0,:);
    stParameters.P_0 = P_0;

    stParameters.alphaV = .0017; % Rate of translation down the "optimal" HC
    ratio: [0, 0.1];
    stParameters.alphaH = stParameters.alphaV; % Rate of translation down the
    "optimal" HC ratio: [0, 0.1];
    stParameters.betaV = 19.3; % Rate of movement toward the "optimal" ratio:
    [0 100];
    stParameters.betaH = 9.0; % Rate of movement toward the "optimal" ratio:
    [0 100];
    stParameters.v50 = 77.9; % 10.9*stParameters.vStar; % Volume where
    P(clearance) = 50%; [0, 0.91MCM];
    stParameters.Dv = 2.6; % 3.2774; [0 100];
    stParameters.Dh = 0.0015; % 0.793; [0 100];
    stParameters.nSlope = 1;
    stParameters.alphaDecay = NaN;
    stParameters.sOptimalTrend = 'linear';

```

【 15 - 21 】

```

else
    sODEC = 'microcytosis';
    s = 1;
    conn = database(sODEC, '', '');
    csMP = getModelParameters(conn, nModelParametersID);

    ssID = csMP(2);
    csSS = getSteadyStatebyID(conn, ssID);

    ssVersion = csSS(15);
    if (~isfinite(ssVersion))
        ssVersion = 0;
    end

    aFID = csSS(3);
    % Get the advia ID from the sample ID.
    csAdviaFile = getAdviaFileByID(conn, aFID);
    csAdviaFile = expandCSFileNames(csAdviaFile);
    nSampleID = aFID;

    simID = csMP(3);
    csSimulationParameters = getSimulationParametersByID(conn, simID);
    close(conn);

    stParameters.nMeshWidth = csSimulationParameters(5);
    stParameters.nMinVolume = csSimulationParameters(7);
    stParameters.nMaxVolume = csSimulationParameters(6);
    stParameters.nMaxHb = csSimulationParameters(9);
    stParameters.nMinHb = csSimulationParameters(8);
    [V,H] = meshgrid([stParameters.nMinVolume:stParameters.nMeshWidth:stParameters.nMaxVolume
]; ...
    [stParameters.nMinHb:stParameters.nMeshWidth:stParameters.nMaxHb]);
    H = flipud(H);
    stParameters.V = V;
    stParameters.H = H;

    stParameters.sOptimizer = csSimulationParameters(3);
    stParameters.sFunctionalForm = csSimulationParameters(11);
    stParameters.bMatchParameters = 0;
    stParameters.nMinAlpha = csSimulationParameters(13);
    stParameters.nMaxAlpha = csSimulationParameters(14);
    stParameters.sDeathFunction = csSimulationParameters(15);
    stParameters.bParallel = 0;
    if (strcmp(csSimulationParameters(16), 'null'))
        stParameters.sOptimalTrend = 'Linear';
    else
        stParameters.sOptimalTrend = csSimulationParameters(16);
    end
    stParameters.bPenalizeFalsePositives = csSimulationParameters(18);
    if (strcmp(csSimulationParameters(18), 'null') ||
    isnan(csSimulationParameters(18)))
        stParameters.bPenalizeFalsePositives = 0;
    end

    stParameters.nMeanMarrowVolume = csSS(10);
    stParameters.nCVMarrowVolume = csSS(11);
    stParameters.nMeanMarrowHb = csSS(12);
    stParameters.nCVMarrowHb = csSS(13);
    stParameters.nMarrowRho = csSS(14);

```

【 15 - 22 】

```

stParameters.vStar = csSS(4);
stParameters.hStar = csSS(6);
stParameters.nDays = 0;

stParameters.muAgeTarg = csMP(11);

stParameters.alphaV = csMP(12); % Rate of translation down the "optimal"
HC ratio: [0, 0.1];
stParameters.alphaH = csMP(13); % Rate of translation down the "optimal"
HC ratio: [0, 0.1];
stParameters.betaV = csMP(14); % Rate of movement toward the "optimal"
ratio: [0 100];
stParameters.betaH = csMP(15); % Rate of movement toward the "optimal"
ratio: [0 100];
stParameters.v50 = csMP(16); % 10.9*stParameters.vStar; % Volume where
P(clearance) = 50%; [0, 0.91MCM];
stParameters.Dv = csMP(17); % 3.2774; [0 100];
stParameters.Dh = csMP(18); % 0.793; [0 100];
if (size(csMP, 2) == 26)
    stParameters.nSlope = NaN;
else if (size(csMP, 2) == 27)
    stParameters.nSlope = csMP(23);
else if (size(csMP, 2) == 28)
    stParameters.alphaDecay = NaN;
end
stParameters.nSlope = csMP(27);
stParameters.alphaDecay = csMP(28);
end
if (isnan(stParameters.nSlope))
    stParameters.nSlope = 1;
end
if (isnan(stParameters.alphaDecay))
    stParameters.alphaDecay = 1;
end

if, nMCM, nRDM, nMCM, nRDM, nRho, vData, hCData] =
estimateBirthFraction(csAdviaFile(6), ssVersion);
[Phat, P, v, h] = mvnNormalCalcRPT(vData, hCData);
P_0 = interp2(v, h, Phat, V, H, 'linear', 0);
P_0 = P_0./sum(P_0,:);
stParameters.P_0 = P_0;

[nMCM, nRDM, nMCM, nRDM, nRho, vData, hCData] =
compareEstimators(csAdviaFile(5), ssVersion);
stParameters.vData = vData;
stParameters.hCData = hCData;

if (strcmp(stParameters.sOptimalTrend, 'Quasi-linear'))
    stParameters.sQuasiLinearFit = getQuasiLinearFit(stParameters,
stParameters.vData, stParameters.hCData);
end

if 0
    sDF = (stParameters.nCVMarrowVolume*stParameters.nMeanMarrowVolume);
    sDR = (stParameters.nCVMarrowHb*stParameters.nMeanMarrowHb);
    nCovar = (stParameters.nMarrowRho*sDF*sDR);
    nSigma = [sDF^2 nCovar; nCovar sDR^2];
    P_0 = mvnpdf([V(:) H(:)], [stParameters.nMeanMarrowVolume
stParameters.nMeanMarrowHb], nSigma);

```

【 15 - 23 】

```

P_0_1 = reshape(sum(P_0(:))*P_0_1/sum(P_0_1(:), size(V));
end
end
function [r, mu_v, cv_v, mu_h, cv_h] = computeRBCorr(P, V, H)
mu_v = P*(V(:))/sum(P(:));
var_v = P*(V(:).^2)/sum(P(:)) - mu_v^2;
cv_v = 100*sqrt(var_v)/mu_v;
mu_h = P*(H(:))/sum(P(:));
var_h = P*(H(:).^2)/sum(P(:)) - mu_h^2;
cv_h = 100*sqrt(var_h)/mu_h;

cov_vh = P*(V(:).*H(:))/sum(P(:)) - 2*mu_v*mu_h + mu_v*mu_h;
r = cov_vh/sqrt(var_v)/sqrt(var_h);
end

function [mBirthAlive, vMeanAge, Pf, Pt] = computeAgeDistribution(P)
vDailyTotals = sum(P');
ixInsignificant = find(vDailyTotals < 1e-6);
%vDailyTotals(ixInsignificant) = 0;
nDays = numel(vDailyTotals);

% Matrix with current day in column and day-cohort in row.
% The (i,j) entry is the fraction of cells born on the ith day alive on
% the jth day.
mBirthAlive = zeros(nDays, nDays);
for ix = 1:nDays
    mBirthAlive(ix, ix:end) = vDailyTotals(1:(end - ix + 1));
end

% Compute the total density born on each day by adding up the amount
% that each previous cohort lost the previous day.
% mAge(i,j) = the absolute density of cells born on day i that are
% cleared on day j.
mAge = -1*zeros(nDays, 1), diff(mBirthAlive');
vWeights = ones(nDays, 1);
for ix = 2:nDays
    mAge(ix,ix) = 0;
    vWeights(ix) = sum(mAge(:, ix));
    mAge(ix, :) = mAge(ix, :)*vWeights(ix);
end

% Convert mBirthAlive to absolute densities:
% mBirthAlive(i,j) = absolute density of cells born on day i that
% are alive on day j. Column j represents the age distribution
% for circulating cells on day j.
mBirthAlive = mBirthAlive.*zpmat(vWeights, 1, nDays);
for ix = 1:nDays
    vMeanAge(ix) = (mBirthAlive(:, ix))*[ix-1:(ix - nDays + 1)];
end

% Compute the total distribution on the last day.
% The ith cohort has grown for (nDays - i) days.
mWeights = flipud(zpmat(mBirthAlive(:, end), [1 size(P, 2)]));
Pnorm = P.*zpmat(sum(P'), [1 size(P, 2)]);
Pf = Pnorm.*mWeights;
if (size(Pf, 1) > 1)
    Pf = sum(Pf);

```

【 15 - 2 4 】

```

end
Pt = Mat('ones(nDays, size(P, 2));
if (argout == 4)
    tSamples = 1;
    for ix = 1:tSamples:nDays - 1
        nWeights = floor(size(nDays)/nDays*(ix, ix), 1) * size(P, 2));
        Ptemp = P * ones(1, nWeights);
        if (size(Ptemp, 1) > 1)
            Ptemp = sum(Ptemp);
        end
        Pt(max(1, floor(ix/tSamples)), :) = Ptemp;
    end
Pt(end, :) = P;
end
end

function [P, t, V, H, stParameters] = fDRBC(stParameters)
P = [];
t = [];
% Get the parameters.
if ~exist('stParameters')
    stParameters = gDRBCDFParameters();
end
nDays = 10;
V = stParameters.V;
H = stParameters.H;
P_D = stParameters.P;

% Setup the clearance function.
pCleared = getClearance(stParameters, V, H);
stParameters.pCleared = pCleared;

% Set up the indices of the mesh points on the boundary.
ixTopRight = (numel(V) - size(V, 1) + 1);
ixBottomRight = numel(V);
ixBottomLeft = size(V, 1);
ixTopLeft = 1;
ixRightEdge = [(numel(V) - size(V, 1) + 2):(numel(V) - 1)];
ixTopEdge = [size(V, 2) + 1:size(V, 1) + 2*size(V, 1) + 1];
ixLeftEdge = [2:size(V, 1) - 1];
ixBottomEdge = [2*size(V, 1):size(V, 1):(numel(V) - size(V, 1))];
ixVStep = size(V, 1);
ixInterior = setdiff(1:(numel(V)), [ixTopRight ixBottomRight ixBottomLeft ixTopLeft ixRightEdge ixTopEdge ixLeftEdge ixBottomEdge]);
mVDot = vdot(stParameters, V, H);
mDot = hdot(stParameters, V, H);

if (0)
    % Code to draw the velocity fields.
    clf
    nDays = 0.25;
    [V, H] = meshgrid(stParameters.mMinVolume:1:stParameters.mMaxVolume, ...
        [stParameters.mMinH:0.1:stParameters.mMaxH]);
    mVDot = vdot(stParameters, V, H);
    mDot = hdot(stParameters, V, H);
    VC = H./V;
end

```

【 15 - 2 6 】

```

function pCleared = getClearance(stParameters, V, H)
nSlope = stParameters.nSlope;
if (isfield(stParameters, 'MeshFunction'))
    assert(strcmp(stParameters.sDeathFunction, 'Projection') ||
        strcmp(stParameters.sDeathFunction, 'Dimensionless Projection') || ...
        strcmp(stParameters.sDeathFunction, 'Tunable Dimensionless Projection') ||
        strcmp(stParameters.sDeathFunction, '0.1 Dimensionless Projection'));
else
    assert(strcmp(stParameters.sDeathFunction, '0.1 Dimensionless Projection'));
end
if (strcmp(stParameters.sDeathFunction, '0.1 Dimensionless Projection'))
    nSlope = 0.1;
    stParameters.nSlope = 0.1;
end
if (strcmp(stParameters.sOptimalTrend, 'Quasi-linear'))
    theta = atan(stParameters.qQuasiLinearFit.a);
    mAngle = atan((H - stParameters.qQuasiLinearFit.b)/V);
    mSlope = (V.^2 + (H - stParameters.qQuasiLinearFit.b).^2).^(1/2);
    vs50 = (stParameters.v50)^2 +
        (stParameters.v50*stParameters.qQuasiLinearFit.a)^2)^(1/2);
    else
        theta = atan(stParameters.hStar/stParameters.vStar);
        mAngle = atan(H./V);
        mSlope = (V.^2 + H.^2).^(1/2);
        vs50 = (stParameters.v50)^2 +
            (stParameters.v50*stParameters.hStar/stParameters.vStar)^2)^(1/2);
        end
        Vz = mSlope*cos(theta - mAngle);
    if (strcmp(stParameters.sFunctionalForm, 'Hybrid') || ...
        strcmp(stParameters.sFunctionalForm, 'Exponential') || ...
        strcmp(stParameters.sFunctionalForm, 'Hybrid Additive Exponential') || ...
        strcmp(stParameters.sFunctionalForm, 'Decaying Additive Exponential') || ...
        strcmp(stParameters.sFunctionalForm, 'Additive Exponential'))
        if (strcmp(stParameters.sFunctionalForm, 'Exponential') || ...
            strcmp(stParameters.sFunctionalForm, 'Hybrid Additive Exponential') || ...
            strcmp(stParameters.sFunctionalForm, 'Decaying Additive Exponential'))
            pCleared = 1./(1 + exp(nSlope*100*(Vz - vs50)/vs50));
        else
            pCleared = 1./(1 + exp(nSlope*(Vz - vs50)));
        end
    elseif (strcmp(stParameters.sFunctionalForm, 'Simple'))
        pCleared = (Vz < vs50);
    end
    pCleared = reshape(pCleared, size(V));
else
    theta = atan(stParameters.hStar/stParameters.vStar);
    R = cos(theta) * sin(theta) - 1 * sin(theta) * cos(theta);
    VC = R ./ (V.^2 + H.^2);
    vs50 = R(1) * stParameters.v50;
    stParameters.v50 = stParameters.hStar/stParameters.vStar;
    if (strcmp(stParameters.sFunctionalForm, 'Hybrid') ||
        strcmp(stParameters.sFunctionalForm, 'Exponential') || ...
        strcmp(stParameters.sFunctionalForm, 'Hybrid Additive Exponential') || ...
        strcmp(stParameters.sFunctionalForm, 'Additive Exponential'))
        stmp = strcmp(stParameters.sFunctionalForm, 'Decaying Additive Exponential');
        pCleared = 1./(1 + exp(nSlope*100*(Vz - vs50)/vs50));
    else
        stmp = strcmp(stParameters.sFunctionalForm, 'Simple');
        pCleared = (Vz < vs50);
    end
    pCleared = reshape(pCleared, size(V));
end

```

【 15 - 2 5 】

```

ix = find(VC(:) > 1.0 -
nRange) * (stParameters.hStar/stParameters.vStar);
ix = intersect(ix, find(VC(:) < 1.0 +
nRange) * (stParameters.hStar/stParameters.vStar));
nLength = nVDot.^2 + mDot.^2;
nScale = 4;
hVz = quiver(V(ix), H(ix), -1*(abs(mVDot(ix))/nScale), -
1*(abs(mDot(ix))/nScale), 'autoscalefactor', 1, 'linewidth', 2, 'maxheadsiz',
0.5);
plot([0 2*stParameters.vStar], [0 2*stParameters.hStar], 'color', 'k',
'linewidth', 4);
hold on;
gr = quiver(V(ix), H(ix), -1*(abs(mVDot(ix))/nScale), -
1*(abs(mDot(ix))/nScale), 'autoscale', 'off', 'linewidth', 2, 'maxheadsiz', 0.2,
'color', 'b');
vL = [29.5 30.5];
sL = [27.5 32.5];
sL = (stParameters.vStar/stParameters.hStar)*sL;
set(gca, 'ylin', 'y');
set(gca, 'xlin', 'x');
nSize = 20;
xlabel('Volume [L]', 'fontsize', nSize);
ylabel('Pressure [Pa]', 'fontsize', nSize);
set(gca, 'fontsize', nSize);
end

[spJacobian, J] = getJacobian(stParameters, mVDot, mDot, ...
ixTopRight, ixBottomRight, ixBottomLeft, ixTopLeft, ixRightEdge,
ixTopEdge, ixLeftEdge, ixBottomEdge, ixInterior, ixVStep);
stParameters.J = J;

[spJacobian, J] = getJacobian(stParameters, numel(mVDot), ixTopRight,
ixBottomRight, ixBottomLeft, ixTopLeft, ...
ixRightEdge, ixTopEdge, ixLeftEdge, ixBottomEdge, ixInterior, ixVStep);
spJacobian = spzeros('l', ...
stParameters.nL);
if (stParameters.nDays)
    odOptions = odset('Vectorized', 'on', 'JPattern', spJacobian,
'OutputFcn', 'odeplot');
else
    odOptions = odset('Vectorized', 'on', 'JPattern', spJacobian, 'AbsTol',
1e-6, 'OutputFcn', 'odeplot');
end
options = odset('Vectorized', 'on', 'JPattern', spJacobian);
opts = optimset('TolX', 10^(-12), 'MaxFun', 1e10, 'MaxIter', 1e10);
[t, P] = fcnbvp(@f, @bc, P_0, opts);
[t, P] = fminsearch(@f, P_0, opts);
end

function df_dt = fDPwind(t, P)
df_dt = zeros(size(P), 1), size(P, 2));
[dp_Cleared, mCleared] = clearRBC(pCleared, P, V, H);
P_burn = P_D * mCleared;
if (0)
    df_dt = J*P + L*P - dp_Cleared + P_burn;
else
    df_dt = J*P + L*P - dp_Cleared;
end
end

```

【 15 - 2 7 】

```

stmp = strcmp(stParameters.sFunctionalForm, 'Decaying Additive Exponential');
pCleared = 1./(1 + exp(nSlope*100*(Vz - vs50)/vs50));
elseif (strcmp(stParameters.sFunctionalForm, 'Simple'))
    pCleared = (Vz < vs50);
end
pCleared = reshape(pCleared, size(V));
end

function [dp_Cleared, mCleared] = clearRBC(pCleared, P, V, H)
% The probability of being cleared is sigmoidal (logistic).
pCleared = sigmoid(pCleared);
% Keep track of how many (probability density) are cleared.
dp_Cleared = sum(dp_Cleared);
end

function [spJacobian, J] = getJacobian(stParameters, mVz, ixTopRight,
ixBottomRight, ixBottomLeft, ixTopLeft, ...
ixRightEdge, ixTopEdge, ixLeftEdge, ixBottomEdge, ixInterior, ixVStep)
Lv = -2*apeye(nVz);
Lv(ixTopRight, [ixTopRight - ixVStep]) = 1;
Lv(ixBottomRight, [ixBottomRight - ixVStep]) = 1;
Lv(ixBottomLeft, [ixBottomLeft + ixVStep]) = 1;
Lv(ixTopLeft, [ixTopLeft + ixVStep]) = 1;
for ix = 1:length(ixRightEdge)
    Lv(ixRightEdge(ix), [ixRightEdge(ix) - ixVStep]) = 1;
end
for ix = 1:length(ixLeftEdge)
    Lv(ixLeftEdge(ix), [ixLeftEdge(ix) + ixVStep]) = 1;
end
for ix = 1:length(ixBottomEdge)
    Lv(ixBottomEdge(ix), [ixBottomEdge(ix) - ixVStep]) = 1;
end
for ix = 1:length(ixTopEdge)
    Lv(ixTopEdge(ix), [ixTopEdge(ix) + ixVStep]) = 1;
end
for ix = 1:length(ixInterior)
    Lv(ixInterior(ix), [ixInterior(ix) - ixVStep]) = 1;
end
Lv = stParameters.Dv/Lv/(stParameters.nMeshWidth^2);
Ln = -3*apeye(nLn);
Ln(ixTopRight, [ixTopRight + 1]) = 1;
Ln(ixBottomRight, [ixBottomRight - 1]) = 1;
Ln(ixBottomLeft, [ixBottomLeft - 1]) = 1;
Ln(ixTopLeft, [ixTopLeft + 1]) = 1;
for ix = 1:length(ixRightEdge)
    Ln(ixRightEdge(ix), [ixRightEdge(ix) + 1]) = 1;
end
for ix = 1:length(ixTopEdge)
    Ln(ixTopEdge(ix), [ixTopEdge(ix) + 1]) = 1;
end

```

【 15 - 2 8 】

```

end
for ix = 1:length(ixLeftEdge)
    Lh(ixLeftEdge(ix), [(ixLeftEdge(ix) + 1) (ixLeftEdge(ix) - 1)]) = 1;
end
for ix = 1:length(ixBottomEdge)
    Lh(ixBottomEdge(ix), [(ixBottomEdge(ix) - 1)]) = 1;
end
for ix = 1:length(ixInterior)
    Lh(ixInterior(ix), [(ixInterior(ix) + 1) (ixInterior(ix) - 1)]) = 1;
end
Lh = stParameters.Dh*Lh/(stParameters.nMeshWidth^2);

L = Lv + Lh;
spLaplacian = spones(L);
end

function [spJacobian, J] = getJacobian(stParameters, mVDot, mHDot, ...
    ixTopRight, ixBottomRight, ixBottomLeft, ixTopLeft, ixRightEdge,
    ixTopEdge, ixLeftEdge, ixBottomEdge, ixInterior, ixVStep)

Jv = -1*speye(numel(mVDot));
Jh = -1*speye(numel(mHDot));

%% Jv
%Jv(ixTopRight, []);
%Jv(ixBottomRight, []);
Jv(ixBottomLeft, [(ixBottomLeft + ixVStep) - 1];
Jv(ixTopLeft, [(ixTopLeft + ixVStep)] = 1;
% Jv(ixRightEdge, []);
for ix = 1:length(ixTopEdge)
    Jv(ixTopEdge(ix), [(ixTopEdge(ix) + ixVStep)] - 1;
end
for ix = 1:length(ixLeftEdge)
    Jv(ixLeftEdge(ix), [(ixLeftEdge(ix) + ixVStep)] = 1;
end
for ix = 1:length(ixBottomEdge)
    Jv(ixBottomEdge(ix), [(ixBottomEdge(ix) + ixVStep)] = 1;
end
for ix = 1:length(ixInterior)
    Jv(ixInterior(ix), [(ixInterior(ix) + ixVStep)] = 1;
end
Jv = Jv.*(repmat(mVDot(:)', size(Jv, 1), 1));
Jv = Jv/stParameters.nMeshWidth;

%% Jh
%Jh(ixTopRight, []);
%Jh(ixTopLeft, []);
Jh(ixBottomLeft, [(ixBottomLeft - 1)] = 1;
Jh(ixBottomRight, [(ixBottomRight - 1)] = 1;
%Jh(ixTopEdge, []);
for ix = 1:length(ixRightEdge)
    Jh(ixRightEdge(ix), [(ixRightEdge(ix) - 1)] = 1;
end
for ix = 1:length(ixLeftEdge)
    Jh(ixLeftEdge(ix), [(ixLeftEdge(ix) - 1)] = 1;
end
for ix = 1:length(ixBottomEdge)
    Jh(ixBottomEdge(ix), [(ixBottomEdge(ix) - 1)] = 1;
end
for ix = 1:length(ixInterior)
    Jh(ixInterior(ix), [(ixInterior(ix) - 1)] = 1;

```

【 15 - 3 0 】

```

deltah = max(0, dh);
nFast = min(zeros(size(deltah)), deltah -
stParameters.nCMMarrowDB/(stParameters.betah);
nSlow = max(deltah, stParameters.nCMMarrowDB/(stParameters.betah)/4;
dh_dh = -1*stParameters.alpha*(stParameters.vStar/stParameters.hStar)^(1
+ nFast + nSlow);
elseif (strcmp(stParameters.sFunctionalForm, 'Simple'))
    dh_dh = -1*stParameters.alpha*(stParameters.hStar)...
    max(1, stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Additive Exponential'))
    dh_dh = -1*stParameters.alpha*(stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Decaying Additive
Exponential'))
    nSlow = (V.*2 + H.*2)^(1/2);
starNorm = sqrt(stParameters.vStar^2 + stParameters.hStar^2);
starNorm/(starNorm - stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Hybrid Exponential'))
    dh_dh = -1*stParameters.alpha*(stParameters.betah*(dh));
    exp(stParameters.betah*(dh));
end
end

function dh = getHDistance(stParameters, V, H)
dh = getHDistance(stParameters, V, H);
if (strcmp(stParameters.sFunctionalForm, 'Hybrid'))
    dh_dh = -1*stParameters.alpha*(stParameters.hStar)...
    exp(stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Exponential'))
    dh_dh = -1*stParameters.alpha*(stParameters.hStar)...
    max(1, stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Piecewise'))
    nSlow = (V.*2 + H.*2)^(1/2);
starNorm = sqrt(stParameters.vStar^2 + stParameters.hStar^2);
starNorm/(starNorm - stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Hybrid Exponential'))
    dh_dh = -1*stParameters.alpha*(stParameters.betah*(dh));
    exp(stParameters.betah*(dh));
end
end

function [x, resnorm, output, jacobian, stParameters] = optPDFDF(vData, hcData,
stParameters)
if (is nargin == 2)
    stParameters = getPDFDFParameters();
end
bMatchParameters = stParameters.bMatchParameters;
bNonlinearPositive = stParameters.bNonlinearPositive;

% Use an empirical target distribution.
[rhat, v_empirical, va, h] = mkEmpiricalPDF(vData, hcData);
x0Cells = find(rhat > 5);
nCells = sum(rhat(x0Cells));
stParameters.nCells = numel(x0Cells);
[x_targ, mu_v_targ, sd_v_targ, mu_h_targ, hdw_targ] = computeSBObs(rhat(:),
va, h);
mu_v_targ = stParameters.muAgeTarg;
vTarg = [v_targ, ...
mu_v_targ, ...
sd_v_targ, ...
hdw_targ, ...
mu_h_targ];
if (strcmp(stParameters.sOptimalTrend, 'Quasi-linear'))
    x0 = [stParameters.alpha, ...
stParameters.betah, ...

```

【 15 - 2 9 】

```

end
Jh = Jh.*(repmat(mHDot(:)', size(Jh, 1), 1));
Jh = Jh/stParameters.nMeshWidth;

J = Jv + Jh;
J = -1*J;

spJacobian = spones(J);
end

function dh_dh = vdot(stParameters, V, H)
dh = getHDistance(stParameters, V, H);
if (strcmp(stParameters.sFunctionalForm, 'Hybrid'))
    dh_dh = -1*stParameters.alpha*(stParameters.hStar)...
    exp(stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Exponential'))
    dh_dh = -1*stParameters.alpha*(stParameters.hStar)...
    max(1, stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Piecewise'))
    nSlow = max(deltah, stParameters.nCMMarrowVolume/(stParameters.betah)/4;
    dh_dh = -1*stParameters.alpha*(stParameters.vStar/(stParameters.hStar)^(1
+ nFast + nSlow));
elseif (strcmp(stParameters.sFunctionalForm, 'Simple'))
    dh_dh = -1*stParameters.alpha*(stParameters.vStar)...
    max(1, stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Additive Exponential'))
    dh_dh = -1*stParameters.alpha*(stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Decaying Additive
Exponential'))
    nSlow = (V.*2 + H.*2)^(1/2);
starNorm = sqrt(stParameters.vStar^2 + stParameters.hStar^2);
starNorm/(starNorm - stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Hybrid Exponential'))
    dh_dh = -1*stParameters.alpha*(stParameters.betah*(dh));
    exp(stParameters.betah*(dh));
end
end

function dh = vdot(stParameters, V, H)
dh = getHDistance(stParameters, V, H);
if (strcmp(stParameters.sOptimalTrend, 'Quasi-linear'))
    a = stParameters.oQuasiLinearFit.a;
    b = stParameters.oQuasiLinearFit.b;
    dh = (V - (H + b)/a)/stParameters.vStar;
else
    dh = V/stParameters.vStar - H/stParameters.hStar;
end
end

function dh_dh = hdot(stParameters, V, H)
dh = getHDistance(stParameters, V, H);
if (strcmp(stParameters.sFunctionalForm, 'Hybrid'))
    dh_dh = -1*stParameters.alpha*(stParameters.hStar)...
    exp(stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Exponential'))
    dh_dh = -1*stParameters.alpha*(stParameters.hStar)...
    max(1, stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Piecewise'))
    nSlow = (V.*2 + H.*2)^(1/2);
starNorm = sqrt(stParameters.vStar^2 + stParameters.hStar^2);
starNorm/(starNorm - stParameters.betah*(dh));
elseif (strcmp(stParameters.sFunctionalForm, 'Hybrid Exponential'))
    dh_dh = -1*stParameters.alpha*(stParameters.betah*(dh));
    exp(stParameters.betah*(dh));
end
end

function [x0, resnorm, output, jacobian, stParameters] =
    optPDFDF(vData, hcData, stParameters)
x0 = [x0, stParameters.alpha, ...
stParameters.v50, ...
stParameters.Dv, ...
stParameters.Dh, ...
stParameters.alpha];
else
    x0 = [stParameters.alpha, ...
stParameters.betah, ...
stParameters.v50, ...
stParameters.Dv, ...
stParameters.Dh];
end

if (strcmp(stParameters.sDeathFunction, 'Tunable Dimensionless Projection'))
    x0 = [x0, stParameters.nSlope];
end
end

if (strcmp(stParameters.sFunctionalForm, 'Decaying Additive Exponential'))
    x0 = [x0, stParameters.alphaDecay];
end
end

resnorm = [];
jacobian = [];
lb = zeros(size(x0));
ub = inf(size(x0));
if (~isnan(stParameters.nMinAlpha))
    lb(1) = stParameters.nMinAlpha;
    ub(1) = stParameters.nMaxAlpha;
end

if (strcmp(stParameters.sOptimizer, 'lsqnonlin'))
    optset = optimset('Display', 'iter', 'TolFun', 1e-15, 'TolX', 1e-15);
    [x, resnorm, residual, exitflag, output, lambda, jacobian] =
lsqnonlin(@nestedfun, x0, lb, ub, optset);
elseif (strcmp(stParameters.sOptimizer, 'patternsearch'))
    if (stParameters.bParallel)
        optset = psoptimset('Display', 'diagnose', 'CompletePoll', 'on',
'UseParallel', 'always');
    else
        optset = psoptimset('Display', 'diagnose');
    end
    [x, resnorm, exitflag, output] = patternsearch(@nestedfun, x0, [], [],
[], [], lb, ub, [], optset);
else
    optset = optimset('Display', 'iter', 'MaxFunEvals', 1000, 'TolFun', 1e-
2);
    [x, resnorm, exitflag, output] = fminsearch(@nestedfun, x0, optset);
end

% Nested function that computes the objective function
function y = nestedfun(x)

% Setup the parameters.
disp(x);
stParameters.alpha = x(1); % Rate of translation down the "optimal" HC
ratio.
bQuasiLinear = strcmp(stParameters.sOptimalTrend, 'Quasi-linear');
if (bQuasiLinear)
    stParameters.alpha = x(7); % Rate of translation down the "optimal"
HC ratio.
else

```

【 15 - 3 1 】

```

stParameters.betah, ...
stParameters.v50, ...
stParameters.Dv, ...
stParameters.Dh, ...
stParameters.alpha];
else
    x0 = [stParameters.alpha, ...
stParameters.betah, ...
stParameters.v50, ...
stParameters.Dv, ...
stParameters.Dh];
end

if (strcmp(stParameters.sDeathFunction, 'Tunable Dimensionless Projection'))
    x0 = [x0, stParameters.nSlope];
end
end

if (strcmp(stParameters.sFunctionalForm, 'Decaying Additive Exponential'))
    x0 = [x0, stParameters.alphaDecay];
end
end

resnorm = [];
jacobian = [];
lb = zeros(size(x0));
ub = inf(size(x0));
if (~isnan(stParameters.nMinAlpha))
    lb(1) = stParameters.nMinAlpha;
    ub(1) = stParameters.nMaxAlpha;
end

if (strcmp(stParameters.sOptimizer, 'lsqnonlin'))
    optset = optimset('Display', 'iter', 'TolFun', 1e-15, 'TolX', 1e-15);
    [x, resnorm, residual, exitflag, output, lambda, jacobian] =
lsqnonlin(@nestedfun, x0, lb, ub, optset);
elseif (strcmp(stParameters.sOptimizer, 'patternsearch'))
    if (stParameters.bParallel)
        optset = psoptimset('Display', 'diagnose', 'CompletePoll', 'on',
'UseParallel', 'always');
    else
        optset = psoptimset('Display', 'diagnose');
    end
    [x, resnorm, exitflag, output] = patternsearch(@nestedfun, x0, [], [],
[], [], lb, ub, [], optset);
else
    optset = optimset('Display', 'iter', 'MaxFunEvals', 1000, 'TolFun', 1e-
2);
    [x, resnorm, exitflag, output] = fminsearch(@nestedfun, x0, optset);
end

% Nested function that computes the objective function
function y = nestedfun(x)

% Setup the parameters.
disp(x);
stParameters.alpha = x(1); % Rate of translation down the "optimal" HC
ratio.
bQuasiLinear = strcmp(stParameters.sOptimalTrend, 'Quasi-linear');
if (bQuasiLinear)
    stParameters.alpha = x(7); % Rate of translation down the "optimal"
HC ratio.
else

```


フロントページの続き

(51)Int.Cl.		F I
A 6 1 K	31/555 (2006.01)	A 6 1 K 31/7135
A 6 1 K	31/295 (2006.01)	A 6 1 K 31/555
A 6 1 K	9/20 (2006.01)	A 6 1 K 31/295
A 6 1 P	3/12 (2006.01)	A 6 1 K 9/20
A 6 1 P	7/06 (2006.01)	A 6 1 P 3/12
		A 6 1 P 7/06

審査官 藤田 都志行

- (56)参考文献 特表2009-524068(JP,A)
 特開平11-326315(JP,A)
 特開平07-105166(JP,A)
 特開2009-036587(JP,A)
 特表2009-524069(JP,A)
 特表2010-526873(JP,A)
 特表2009-510402(JP,A)
 米国特許第06030838(US,A)
 小澤 敬也,「貧血の病態と診断・治療」,日本臨牀,株式会社日本臨牀社,2008年 3月
 1日,Vol. 66, No. 3, pp. 423-428

(58)調査した分野(Int.Cl., DB名)

G 0 1 N 3 3 / 4 9
 G 0 1 N 3 3 / 7 2
 A 6 1 K 9 / 2 0
 A 6 1 K 3 1 / 2 9 5
 A 6 1 K 3 1 / 5 5 5
 A 6 1 K 3 1 / 7 1 3 5
 A 6 1 K 3 3 / 2 6
 A 6 1 K 3 8 / 2 2
 A 6 1 P 3 / 1 2
 A 6 1 P 7 / 0 6
 J S T P l u s / J M E D P l u s / J S T 7 5 8 0 (J D r e a m I I I)
 医中誌WEB