(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103430028 A
(43)申请公布日 2013.12.04

(21)申请号 201180055278.3
(22)申请日 2011.09.16
(30)优先权数据
61/383,357 2010.09.16 US
(85)PCT申请进入国家阶段日
2013.05.16
(86)PCT申请的申请数据
PCT/US2011/052038 2011.09.16
(87)PCT申请的公布数据
WO2012/037524 EN 2012.03.22
(71)申请人 通用医疗公司
地址 美国马萨诸塞州
(72)发明人 J.M. 希金斯
(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 张文辉
(51)Int.Cl.
*G01N 33/72*(2006.01)
*G01N 33/555*(2006.01)
*G01N 33/49*(2006.01)

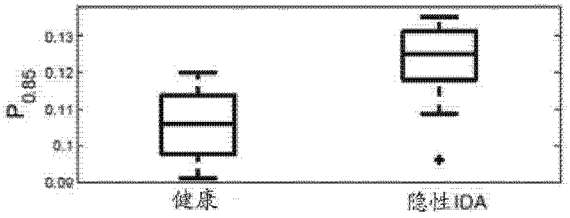
权利要求书5页 说明书21页 附图52页

(54)发明名称

供诊断用的红细胞动力学

(57)摘要

用于鉴定贫血患者、区别珠蛋白生成障碍性贫血性状贫血与铁缺乏贫血、及在贫血变得临床可检出前几周鉴定前贫血患者的方法。还有,用于检测运动员中的血掺以及用于优化用红细胞生成刺激剂或铁补充的疗法的方法。计算机可读存储装置和系统,例如其用于所描述的方法。




CN 103430028 A

1. 一种测定受试者形成铁缺乏贫血 (IDA) 的风险的方法，该方法包括：

在包含来自所述受试者的红细胞的样品中，测定群体平均红细胞血红蛋白浓度 (MCHC)；

将每个细胞的体积和血红蛋白含量转化成指数，其通过将所述体积和血红蛋白含量投影到所述 MCHC 线上来进行；

测定如下样品中的红细胞分数以提供样品分数，所述样品的转化体积和血红蛋白含量指数降到所述 MCHC 线上的均值投影位置的阈值百分比下；并

比较所述样品分数与参照分数；

其中高于所述参照分数的样品分数的存在指示所述受试者处于形成 IDA 的风险。

2. 一种筛选或选择受试者以筛选胃肠 (GI) 病症的方法，该方法包括：

在包含来自受试者的红细胞的样品中，测定群体平均红细胞血红蛋白浓度 (MCHC)；

将每个细胞的体积和血红蛋白含量转化成指数，其通过将所述体积和血红蛋白含量投影到所述 MCHC 线上来进行；

测定如下样品中的红细胞分数，所述样品的转化体积和血红蛋白含量指数降到所述 MCHC 线上的均值投影位置的阈值百分比下；

比较所述样品分数与参照分数；并

在所述样品分数低于所述参照分数时选择受试者以进行 GI 评估或者进一步筛选。

3. 一种在患有小红细胞性贫血的受试者中在铁缺乏贫血 (IDA) 和珠蛋白生成障碍性贫血性状 (TT) 之间做出区别诊断的方法，该方法包括：

在来自受试者的红细胞样品中测定细胞间血红蛋白含量降低 ($D_h$) 速率的变化幅度的数值，并比较 $D_h$ 与参照数值；

其中高于所述参照数值的 $D_h$ 的存在指示所述受试者患有或更有可能患有 IDA，而低于所述参照数值的 $D_h$ 的存在指示所述受试者患有或更有可能患有 TT。

4. 一种用于检测正常受试者中血掺或红细胞生成刺激物药剂的存在或使用的方法，该方法包括：

在来自受试者的样品中测定血红蛋白含量降低 ($D_h$) 速率的变化幅度和细胞体积降低 ($D_v$) 速率的变化幅度中的一项或两项的数值；

其中高于参照数值的 $D_h$ 和 / 或 $D_v$ 的存在指示血掺的存在或使用或者使用红细胞生成刺激物药剂。

5. 一种用于优化受试者中红细胞生成素 (EPO) 或其它红细胞生成刺激物药剂 (ESA) 剂量的方法，该方法包括：

在来自经历 EPO 或 ESA 治疗的受试者的样品中测定标准化临界体积 ($v_c$)；并且

若所述 $v_c$ 低于下参照水平，则提高 EPO 剂量，或者

若所述 $v_c$ 高于上参照水平，则降低 EPO 剂量。

6. 权利要求 5 的方法，其中调节 EPO 的剂量以维持所述受试者中群体均值体积 ($\bar{v}$) 的大于 75% 的 $v_c$；或 78-82% 的 $v_c$。

7. 一种用铁治疗受试者的方法，该方法包括：

在来自受试者的样品中测定标准化临界体积 ($v_c$)；并且

若所述 $v_c$ 低于下参照水平，则施用一剂铁或者规定铁补充过程。

8. 权利要求 7 的方法，其中调节铁的剂量以维持所述受试者中群体均值体积($\bar{v}$)的大于 78% 的 $v_c$；或 78-82% 的 $v_c$。

9. 权利要求 1 或 2 的方法，其中所述阈值百分比是所述 MCHC 线上所述群体均值投影的 70%、75%、80%、85%、90% 或 95%。

10. 权利要求 1 或 2 的方法，其中所述参照分数是约 0.10、0.11、0.12、0.13、0.14、或 0.15。

11. 权利要求 1-10 中任一项的方法，其中所述样品包含来自所述受试者的全血。

12. 一种系统，其包含：

计算装置，该计算装置包含：

用于存储指令的存储器，

一个或多个能够执行存储指令的处理器或处理装置以实施如下的操作，其包括：

接受代表所述样品中每个细胞的体积和血红蛋白浓度或含量的数据；将每个细胞的体积和血红蛋白含量转化成指数，其通过将所述体积和血红蛋白含量投影到代表平均红细胞血红蛋白浓度 (MCHC) 的线上来进行；

计算如下样品中红细胞的分数，所述样品的转化体积和血红蛋白含量指数降到均值投影的阈值百分比下；并

提供代表该分数的输出。

13. 权利要求 12 的系统，其进一步包含接受代表所述样品中平均红细胞血红蛋白浓度 (MCHC) 的数据的指令。

14. 权利要求 12 的系统，其进一步包含检测模块，该检测模块配置为检测红细胞 (RBC) 细胞体积 (CV)、平均细胞体积 (MCV)、细胞血红蛋白浓度 (CHC)、平均细胞血红蛋白浓度 (MCHC)、和平均细胞血红蛋白含量 (MCH) 及其群体统计学中的一项或多项，例如所有。

15. 一个或多个机器可读存储装置，其配置为存储由一个或多个处理器或处理装置可执行以实施如下操作的指令，所述操作包括：

接受代表所述样品中每个细胞中的体积和血红蛋白浓度的数据；将每个细胞的体积和血红蛋白含量转化成指数，其通过将所述体积和血红蛋白含量投影到代表平均红细胞血红蛋白浓度 (MCHC) 的线上来进行；

计算如下样品中红细胞的分数，所述样品的转化体积和血红蛋白含量指数降到均值投影的阈值百分比下；并

提供代表该分数的输出。

16. 权利要求 15 的机器可读存储装置，其进一步包含接受代表所述样品中平均红细胞血红蛋白浓度 (MCHC) 的数据的指令。

17. 权利要求 12-14 中任一项的系统或权利要求 15-16 中任一项的机器可读存储装置，其进一步包含计算下列一项或多项的指令：

(i) 血红蛋白含量降低 ($D_h$) 速率的变化幅度；

(ii) 细胞体积降低 ($D_v$) 速率的变化幅度；

(iii) 标准化临界体积或清除阈值 ($v_c$)；

(iv) 血红蛋白含量降低 (α) 和慢相体积的平均速率；

(v) 快相体积降低 ($\beta_v$) 的平均速率；和

(vi) 快相血红蛋白含量降低 ($\beta_h$) 的平均速率。

18. 一个或多个机器可读存储装置，其包含计算下列一项或多项的机器可读指令：

(i) 群体平均红细胞血红蛋白浓度 (MCHC)、通过将体积和血红蛋白含量投影到 MCHC 线上来将每个细胞的体积和血红蛋白含量转化成指数、和投影降到均值投影的阈值百分比下的样品中的红细胞分数；

(ii) 血红蛋白含量降低 ($D_h$) 速率的变化幅度；

(iii) 细胞体积降低 ($D_v$) 速率的变化幅度；和

(iv) 标准化临界体积或清除阈值 ($v_c$)。

19. 一种对患者筛选珠蛋白生成障碍性贫血性状 (TT) 的方法，该方法包括：

在来自受试者的样品中测定血红蛋白含量降低 ($D_h$) 速率的变化幅度的数值，并

比较 $D_h$ 与参照范围；

其中在所述参照范围外的 $D_h$ 的存在指示所述患者患有或有可能患有 TT 和 / 或若所述 $D_h$ 在所述参照范围外，则选择受试者进行基因型分型或别的测试，或者若所述患者是妊娠的女性，则选择父亲进行筛选。

20. 一种在患有贫血的受试者中在慢性病贫血 (ACD) 和其它贫血原因之间做出区别诊断的方法，该方法包括测定下列一项或多项的数值：

(i) 血红蛋白含量降低 ($D_h$) 速率的变化幅度；和 / 或

(ii) 细胞体积降低 ($D_v$) 速率的变化幅度；和 / 或

(iii) 标准化临界体积或清除阈值 ($v_c$)；和 / 或

(iv) 慢相体积和血红蛋白含量降低 ($\alpha$) 的平均速率；和 / 或

(v) 快相体积降低 ($\beta_v$) 的平均速率；和 / 或

(vi) 快相血红蛋白含量降低 ($\beta_h$) 的平均速率；

比较这些数值与参照范围，其中通过所述参照范围内的数值数目确定 ACD 诊断的概率；比较此概率与阈值；并且若所述概率高于所述阈值，则针对输血疗法选择和 / 或选择不实施铁水平的额外测试。

21. 权利要求 1-18 中任一项的方法，其中转化个别细胞的体积和血红蛋白含量，使得所述体积和血红蛋白含量坐标投影到代表所有体积和血红蛋白含量坐标的最小二乘线性拟合的线上。

22. 一种计算机可读存储装置，其配置为存储计算机可读的指令，该指令在由一个或多个处理器执行时引起如下的操作，该操作包括：

接受与网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布相关的第一组参数；

基于所述第一组参数，评估红细胞的稳态分布；并

计算代表红细胞的评估稳态分布和经验分布之间的不相似程度的目标函数。

23. 权利要求 22 的计算机可读存储装置，其进一步包含如下的指令：

若所述目标函数的数值满足预定的阈值条件，则调节所述第一组参数，以提供第二组参数，使得相应的评估稳态分布和所述经验分布之间的不相似程度降低；并

对所述一个或多个处理器提供所述第二组参数作为第一组参数。

24. 权利要求 22-23 中任一项的计算机可读存储装置，其中所述网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布基于自患者获得的血液样品。

25. 权利要求 22-24 中任一项的计算机可读存储装置，其中基于线性算子和测量的分布状态的组合评估所述稳态分布。

26. 权利要求 25 的计算机可读存储装置，其中所述线性算子包含至少一个雅可比行列式和至少一个拉普拉斯算子。

27. 一种系统，其包含 ：

计算装置，该计算装置包含 ：

用于存储指令的存储器，和

检测模块，其包含一个或多个能够执行存储指令的处理器或处理装置以实施如下的操作，该操作包括 ：

接受与网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布相关的第一组参数 ；

基于所述第一组参数，评估红细胞的稳态分布 ；并

计算代表红细胞的评估稳态分布和经验分布之间的不相似程度的目标函数。

28. 权利要求 27 的系统，其进一步配置为 ：

若所述目标函数的数值满足预定的阈值，则调节所述第一组参数，以提供第二组参数，使得相应的评估稳态分布和所述经验分布之间的不相似程度降低 ；并

对所述一个或多个处理器提供所述第二组参数作为第一组参数。

29. 权利要求 27-28 中任一项的系统，其中所述网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布基于自患者获得的血液样品。

30. 权利要求 27-29 中任一项的系统，其中基于线性算子和测量的分布状态的组合评估所述稳态分布。

31. 权利要求 30 的系统，其中所述线性算子包含至少一个雅可比行列式和至少一个拉普拉斯算子。

32. 一种计算机执行的方法，包括 ：

接受与网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布相关的第一组参数 ；

基于所述第一组参数，评估红细胞的稳态分布 ；并

计算代表红细胞的所述评估稳态分布和经验分布之间的不相似程度的目标函数。

33. 权利要求 32 的方法，其进一步包括 ：

若所述目标函数的数值满足预定的阈值条件，则调节所述第一组参数，以提供第二组参数，使得相应的评估稳态分布和所述经验分布之间的不相似程度降低 ；并

对所述一个或多个处理器提供所述第二组参数作为第一组参数。

34. 权利要求 32-33 中任一项的方法，其中所述网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布基于自患者获得的血液样品。

35. 权利要求 32-34 中任一项的方法，其中基于线性算子和测量的分布状态的组合评估所述稳态分布。

36. 权利要求 35 的方法，其中所述线性算子包含至少一个雅可比行列式和至少一个拉

普拉斯算子。

# 供诊断用的红细胞动力学

[0001] 优先权要求

[0002] 本申请要求2010年9月16日提交的美国临时申请流水号61/383,357的权益，其通过提及完整并入本文。

[0003] 联邦资助的研究或开发

[0004] 本发明是在由国立健康研究所授予的拨款号DK083242和HL091331下得到政府支持完成的。政府具有本发明的某些权利。

## 技术领域

[0005] 本发明涉及用于鉴定贫血或前贫血患者，以及区别珠蛋白生成障碍性贫血性状贫血与铁缺乏贫血和其它贫血原因的方法。也可以使用所述方法来优化治疗，例如，用铁补充剂和/或红细胞生成刺激剂诸如红细胞生成素(EPO)的疗法。还描述了在例如所描述的方法中使用的装置和系统。

[0006] 发明背景

[0007] 控制循环人红细胞(RBC)的数目、大小、血红蛋白浓度、和其它特征的系统是了解不足的。在自骨髓释放后，RBC通过未知的机制而经历体积和总血红蛋白含量两者的降低(Lew VL等(1995)Blood86:334-341;Waugh RE等(1992)Blood79:1351-1358)；在约120天后，响应未知的触发物，它们被除去。

[0008] 发明概述

[0009] 部分基于来自统计物理学的理论和来自医院临床实验室的数据(d'Onofrio G等(1995)Blood85:818-823)，本发明使用用于RBC成熟和清除的主方程模型。该模型精确鉴定贫血患者，并且区别珠蛋白生成障碍性贫血性状贫血与铁缺乏贫血。引人注意地，它还在贫血变得临床上可检出前几周鉴定许多前贫血患者。

[0010] 在一个方面，测定受试者形成铁缺乏贫血(IDA)的风险的方法包括在样品中包含来自受试者的红细胞，测定群体平均红细胞血红蛋白浓度(MCHC)，并通过将体积和血红蛋白含量投影到MCHC线上来将每个细胞的体积和血红蛋白含量转化成指数。该方法还包括测定样品中的红细胞分数(fraction)以提供样品分数，所述样品的转化体积和血红蛋白含量指数降到MCHC线上的均值投影位置的阈值百分比下，并比较所述样品分数与参照分数。高于参照分数的样品分数的存在指示受试者有形成IDA的风险。

[0011] 在另一个方面，筛选或选择受试者以筛选胃肠(GI)病症的方法包括在样品中包含来自受试者的红细胞，测定群体平均红细胞血红蛋白浓度(MCHC)，并通过将体积和血红蛋白含量投影到MCHC线上来将每个细胞的体积和血红蛋白含量转化成指数。该方法还包括测定样品中的红细胞分数，所述样品的转化体积和血红蛋白含量指数降到MCHC线上的均值投影位置的阈值百分比下，并比较所述样品分数与参照分数。所述方法进一步包括选择受试者进行GI评估或者在样品分数低于参照分数时进一步筛选。

[0012] 在另一个方面，公开内容的特征在于一种在患有小红细胞性贫血的受试者，例如，没有慢性疾病的受试者中在铁缺乏贫血(IDA)和珠蛋白生成障碍性贫血性状(TT)之间做

出区别诊断的方法。该方法包括在来自受试者的红细胞样品中测定细胞间血红蛋白含量降低 ($D_h$) 速率的变化幅度的数值，并比较 $D_h$ 与参照数值。高于参照数值的 $D_h$ 的存在指示受试者患有或更有可能患有 IDA，而低于参照数值的 $D_h$ 的存在指示受试者患有或更有可能患有 TT。

[0013] 在另一个方面，用于检测正常受试者中血掺或红细胞生成刺激物药剂的存在或使用的方法包括在来自受试者的样品中测定血红蛋白含量降低 ($D_h$) 速率的变化幅度和细胞体积降低 ($D_v$) 速率的变化幅度中的一项或两项的数值。高于参照数值的 Dh 和 / 或 Dv 的存在指示血掺的存在或使用或者使用红细胞生成刺激物药剂。

[0014] 在另一个方面，用于优化受试者中红细胞生成素 (EPO) 或其它红细胞生成刺激物药剂 (ESA) 剂量的方法包括在来自经历 EPO 或 ESA 治疗的受试者的样品中测定标准化临界体积 ($v_c$)。该方法还包括若 $v_c$ 低于下参照水平，则提高 EPO 剂量，或者若 $v_c$ 高于上 (upper) 参照水平，则降低 EPO 剂量。

[0015] 在另一个方面，用铁治疗受试者的方法包括在来自受试者的样品中测定标准化临界体积 ($v_c$)，并且若 $v_c$ 低于下 (lower) 参照水平，则施用一剂 (a dose of) 铁或者规定铁补充过程。

[0016] 在另一个方面，公开内容的特征在于一种包含计算装置的系统，该计算装置包含用于存储指令的存储器和一个或多个能够执行存储指令的处理器或处理装置。存储指令在执行时实施如下的操作，其包括接受代表样品中每个细胞的体积和血红蛋白浓度或含量的数据，并通过将体积和血红蛋白含量投影到代表平均红细胞血红蛋白浓度 (MCHC) 的线上来将每个细胞的体积和血红蛋白含量转化成指数。该操作还包括计算样品中红细胞的分数，并提供代表该分数的输出，所述样品的转化体积和血红蛋白含量指数降到均值投影的阈值百分比下。

[0017] 在另一个方面，公开内容的特征在于一个或多个机器可读存储装置，其配置为存储由一个或多个处理器或处理装置可执行的指令。该指令在由一个或多个处理器执行时实施如下的操作，该操作包括接受代表样品中每个细胞中的体积和血红蛋白浓度的数据，并通过将体积和血红蛋白含量投影到代表平均红细胞血红蛋白浓度 (MCHC) 的线上来将每个细胞的体积和血红蛋白含量转化成指数。该操作还包括计算样品中红细胞的分数，并提供代表该分数的输出，所述样品的转化体积和血红蛋白含量指数降到均值投影的阈值百分比下。

[0018] 在另一个方面，公开内容的特征在于一个或多个机器可读存储装置，其包含机器可读的指令以计算下列一项或多项：

[0019] (i) 群体平均红细胞血红蛋白浓度 (MCHC)、通过将体积和血红蛋白含量投影到 MCHC 线上来将每个细胞的体积和血红蛋白含量转化成指数、和投影降到均值投影的阈值百分比下的样品中的红细胞分数，(ii) 血红蛋白含量降低 ($D_h$) 速率的变化幅度，(iii) 细胞体积降低 ($D_v$) 速率的变化幅度，及 (iv) 标准化临界体积或清除阈值 ($v_c$)。

[0020] 在另一个方面，对患者筛选珠蛋白生成障碍性贫血性状 (TT) 的方法包括在来自受试者的样品中测定血红蛋白含量降低 ($D_h$) 速率的变化幅度的数值，并比较 $D_h$ 与参照范围。在参照范围外的 $D_h$ 的存在指示患者患有或有可能患有 TT 和 / 或若 $D_h$ 在参照范围外，则选择受试者进行基因型分型或别的测试，或者若患者是妊娠的女性，则选择父亲进行筛选。

[0021] 在另一个方面，公开内容的特征在于在患有贫血（例如，已经诊断为贫血）的受试者中在慢性病贫血 (ACD) 和其它贫血原因之间做出区别诊断的方法。该方法包括测定下列一项或多项的数值：(i) 血红蛋白含量降低 ($D_h$) 速率的变化幅度，(ii) 细胞体积降低 ($D_v$) 速率的变化幅度，(iii) 标准化临界体积或清除阈值 ($v_c$)，(iv) 慢相体积和血红蛋白含量降低（α）的平均速率，(v) 快相体积降低（$\beta_v$）的平均速率，和 (vi) 快相血红蛋白含量降低（$\beta_h$）的平均速率；比较这些数值与参照范围，其中通过参照范围内的数值数目确定 ACD 诊断的概率；比较此概率与阈值；并且若概率高于阈值，则针对输血疗法选择和 / 或选择不实施铁水平的额外测试。

[0022] 在另一个方面，公开内容的特征在于一种计算机可读存储装置，其配置为存储计算机可读的指令，该指令在由一个或多个处理器执行时引起如下的操作，该操作包括接受与网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布相关的第一组参数，并基于第一组参数，评估稳态 (steady state) 分布。该操作还包括计算代表红细胞的评估稳态分布和经验分布之间的不相似程度的目标函数。

[0023] 在另一个方面，公开内容的特征在于包含计算装置的系统。该计算装置包含用于存储指令的存储器，和检测模块，该检测模块包含一个或多个能够执行存储指令以实施各种操作的处理器或处理装置。该操作包括接受与网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布相关的第一组参数，并基于第一组参数，评估红细胞的稳态分布。该操作还包括计算代表红细胞的评估稳态分布和经验分布之间的不相似程度的目标函数。

[0024] 在另一个方面，计算机执行的方法包括接受与网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布相关的第一组参数，并基于第一组参数，评估红细胞的稳态分布。该方法还包括计算代表红细胞的评估稳态分布和经验分布之间的不相似程度的目标函数。

[0025] 上述方法、系统和计算机可读存储装置的实现可以包括下列特征的任何组合。

[0026] 可以调节 EPO 的剂量以维持受试者中群体均值体积（$\bar{v}$）的大于 75% 的 $v_c$；或 78-82% 的 $v_c$。可以调节铁剂量以维持受试者中群体均值体积（$\bar{v}$）的大于 78% 的 $v_c$；或 78-82% 的 $v_c$。阈值百分比可以是 MCHC 线上的群体均值投影的 70%、75%、80%、85%、90%、或 95%。参照分数是约 0.10、0.11、0.12、0.13、0.14、或 0.15。样品可以包括来自受试者的全血。可以接受代表样品中平均红细胞血红蛋白浓度 (MCHC) 的数据。检测模块可以配置为检测红细胞 (RBC) 细胞体积 (CV)、平均细胞体积 (MCV)、细胞血红蛋白浓度 (CHC)、平均细胞血红蛋白浓度 (MCHC)、和平均细胞血红蛋白含量 (MCH) 及其群体统计学中的一项或多项，例如所有。可以转化个别细胞的体积和血红蛋白含量，使得所述体积和血红蛋白含量坐标投影到代表所有体积和血红蛋白含量坐标的最小二乘 (least-squares) 线性拟合的线上。

[0027] 计算机或机器可读存储装置可以包含计算下列一项或多项的指令：(i) 血红蛋白含量降低 ($D_h$) 速率的变化幅度，(ii) 细胞体积降低 ($D_v$) 速率的变化幅度，(iii) 标准化临界体积或清除阈值 ($v_c$)，(iv) 慢相体积和血红蛋白含量降低（α）的平均速率，(v) 快相体积降低（$\beta_v$）的平均速率，和 (vi) 快相血红蛋白含量降低（$\beta_h$）的平均速率。

[0028] 若目标函数的数值满足预定的阈值条件，则可以调节所述第一组参数，以提供第二组参数。第二组参数使得相应的评估稳态分布和所述经验分布之间的不相似程度降低。可以对所述一个或多个处理器提供所述第二组参数作为第一组参数。网织红血球或红细胞的时间依赖性联合体积 - 血红蛋白分布可以基于自患者获得的血液样品。可以基于线性算

子和测量的分布状态的组合评估所述稳态分布。线性算子可以包括至少一个雅可比行列式(Jacobian)和至少一个拉普拉斯算子(Laplacian)。

[0029] 除非另有定义,本文中使用的所有技术和科学术语与本发明所属领域普通技术人员的通常理解具有相同的意义。在本文中描述方法和材料以用于本发明;也可以使用本领域中已知的其它合适的方法和材料。材料、方法和例子仅是例示性的,而并不意图为限制性的。本文中提及的所有出版物、专利申请、专利、序列、数据库条目、和其它参考文献通过提及完整并入。在冲突的情况中,应当以本说明书(包括定义)为准。

[0030] 本发明的其它特征和优点通过以下详细描述和图,以及通过权利要求书会是显而易见的。

[0031] 附图简述

[0032] 图1A和1B是共调节外周循环中平均RBC的体积和血红蛋白的经验测量(1A)和动力学模型(1B)。在图1A中,网织红血球分布在等概率密度等高线中以实线显示,而所有RBC的群体分布以虚线显示。投影到这两幅小图中的原点的对角线代表群体中的平均胞内血红蛋白浓度(MCHC)。位于此线上任何地方的RBC会具有等于MCHC的胞内血红蛋白浓度。快动力学(fast dynamics)(β)首先降低每幅小图右上部显示的典型的大的未成熟网织红血球的体积和血红蛋白。慢动力学(α)然后沿着MCHC线降低体积和血红蛋白。因为生物学过程是本质上有噪声的,所以在体积和血红蛋白降低需要的事件期间的小的随机变化可以引起个体细胞血红蛋白浓度在MCHC线周围漂移,在MCHC线周围随幅度(D)波动,如图1B的插图中显示的,直至在消除细胞时达到临界体积(图1B中的$v_c$)。

[0033] 图2是用于20名健康个体和三种形式的轻度贫血患者:患有慢性病贫血(ACD)的11名、患有珠蛋白生成障碍性贫血性状(TT)的33名、和患有铁缺乏贫血(IDA)的27名的模型参数的一组六幅盒式图(boxplot)。每个盒的上部和底部边缘位于第75和第25百分位数。中值以框内部的水平线标示。垂直线延伸到如下的数据点,该数据点离框的距离小于四分位数间距离的1.5倍。更多极值数据点以加号(+)符号显示。快动力学以β表征,慢动力学以α表征,随机波动以D表征,而清除阈值以$v_c$表征。

[0034] 图3A-C是在4个月后形成IDA的患者的全血液计数(CBC)的等高线图。每幅图显示了封入联合体积-血红蛋白含量概率密度的35%、60%、75%和85%的等高线。来自原点的虚线代表MCHC。圆形显示每个细胞的体积-血红蛋白含量坐标投影到MCHC线上的均值。与MCHC线垂直的短实线标记沿着该线与均值投影的85%对应的位置。图3A显示了在患者呈现出IDA前116天测量的正常CBC。计算的$P_{0.85}$(实心灰色)是正常的。图3B显示了检测IDA后65天和前51天测量的正常CBC。$P_{0.85}$是异常的,尽管CBC是正常的。小图图3C显示了诊断出IDA时的CBC。

[0035] 图3D显示了来自在90天内具有第二正常CBC的患者的20个正常CBC和来自直到90天后诊断出IDA的患者的20个正常CBC的$P_{0.85}$的盒式图。$P_{0.85}$以75%的灵敏度和100%的特异性比实际诊断早多至90天成功预测IDA。关于更多详情,参见下文和实施例3,“预测铁缺乏贫血”。

[0036] 图4是一幅盒式图,其显示了5例患有TT的训练病例(training case)、5例患有IDA的训练病例、28例患有TT的测试病例、和22例患有IDA的测试病例的$D_h$的分布。结果例示本方法区别作为小红细胞性贫血原因的TT和IDA的能力。

[0037] 图 5 是用于计算清除概率的投影距离（Δ）的示意图，如方程式 3 中所描述的。将细胞投影到 MCHC 线上，并且清除概率是沿着此线从投影点到阈值（$v_c$）的距离的函数。

[0038] 图 6A-B 是 3D 图，其显示对健康个体比较拟合的（暗灰色）和经验的（亮灰色）稳态联合体积 - 血红蛋白含量概率分布。图 6A 显示了贯穿 MCHC 线（参见图 1）在垂直面上投影的视图。图 6B 显示了朝原点看的 90 度旋转视图。

[0039] 图 7 是来自超过 200 次分开模拟和拟合优度的优化参数数值（如通过残差平方和目标函数测定）的一组六幅直方图。目标函数的较小数值意味着较好的拟合。图 7 显示了所有模型参数具有定义明确的最佳邻域。关于目标函数，参见方程式 5。

[0040] 图 8A 是一幅盒式图，其显示了图 2 中显示的“健康”和“IDA”患者的 $P_{0.85}$ 的分布。基于这些结果，选择 $P_{0.85}$ 的阈值数值 0.121 以在图 3D 中显示的独立的一组患者的测试中使用。此图是用于鉴定隐性 (latent)IDA 阈值的方法的例子。

[0041] 图 8B-8D 是显示图 2 中显示的稳态健康受试者（每幅图中的顶部小图）和 IDA 受试者（每幅图中的底部小图）的 $P_{0.85}$(8B)、$P_{0.75}$(8C)、和 $P_{0.90}$(8D) 的分布的直方图。

[0042] 图 9A-9C 显示了单个 RBC 体积和血红蛋白含量动力学的三种不同函数形式的速度场 (velocity field)，如表 1A-B 中所列的。黑色对角线显示了等于 MCHC 的恒定血红蛋白浓度。

[0043] 图 10A-B 是 3D 图，其显示了表 1A-B 中描述的两种不同清除函数的清除概率。图 10A，函数形式 A 和 C ；图 10B，函数形式 B。MCHC 线以黑色显示。

[0044] 图 11 是模型参数的一组六幅盒式图，其基于 20 名健康个体的来自表 1A-B 的函数形式 B 和它们对于患有三种轻度贫血形式的患者 ：患有慢性病贫血 (ACD) 的 11 名、患有珠蛋白生成障碍性贫血性状的 33 名、和患有铁缺乏贫血的 27 名而言如何变化。每个盒的上部和下部边缘位于第 75 和第 25 百分位数。中值以框内部的水平线标示。垂直线延伸到如下的数据点，该数据点在离框的四分位数间距离的 1.5 倍内。更多极值数据点以加号 (+) 符号显示。快动力学以 β 表征，慢动力学以 α 表征，随机波动以 D 表征，而清除阈值以 $v_c$ 表征。

[0045] 图 12 是模型参数的一组六幅盒式图，其基于 20 名健康个体的来自表 2 的函数形式 C 和它们对于患有三种轻度贫血形式的患者 ：患有慢性病贫血 (ACD) 的 11 名、患有珠蛋白生成障碍性贫血性状的 33 名、和患有铁缺乏贫血的 27 名而言如何变化。每个盒的上部和下部边缘位于第 75 和第 25 百分位数。中值以框内部的水平线标示。垂直线延伸到如下的数据点，该数据点在离框的四分位数间距离的 1.5 倍内。更多极值数据点以加号 (+) 符号显示。快动力学以 β 表征，慢动力学以 α 表征，随机波动以 D 表征，而清除阈值以 $v_c$ 表征。

[0046] 图 13A-B 是框图，其显示了用于测定样品中细胞分数的例示性方法步骤，依照本发明的一些实施方案，所述样品的体积和血红蛋白含量在 MCHC 线上投影时沿着此线降到均值投影位置的阈值百分比下。

[0047] 图 14 是一幅框图，其显示了在本发明中使用的例示性计算装置。

[0048] 图 15 是一种用于从患者血液样品的体积和血红蛋白含量测量推断模型参数的程序的例示性计算机代码。

[0049] 图 16 是一种用于被调函数 (called function) “ssRBC”的例示性计算机代码，所

述被调函数“ssRBC”使用方程式 4 的数字逼近 (approximation) 来计算稳态 RBC 分布。

[0050] 图 17 是用于程序“calculateCompensation”(其计算受试者血液样品的 $P_{0.85}$) 的例示性计算机代码。此函数也调用其它,但是可以不依赖于参数计算代码运行。

[0051] 发明详述

[0052] 红细胞经由了解不足的过程自循环除去。虽然这些和其它 RBC 成熟和清除过程根本的特定分子和细胞机制是未知的,但是可能的是这些过程在疾病状态中改变,并且这些改变导致患有这些疾病的患者中的各种差异。例如,若特定细胞特征,如体积或血红蛋白浓度在红细胞循环时间的过程期间改变,则可能的是来自患有一些疾病的患者的红细胞群体间此变化的平均速度会与来自没有该疾病的患者的红细胞群体间的此变化的平均速度不同。

[0053] 在健康成年人中,每天从骨髓将约 $2.5x10^{11}$ 个新 RBC 释放到循环中,并且大约相同数目被清除。如此,构成循环群体的细胞不断变化,但是在健康个体 ( 和轻度疾病患者 ) 中,群体的特征是非常稳定的。在临床中,群体特征诸如血液中的细胞体积分数 ( 血细胞比容 )、平均 RBC 体积 (MCV)、RBC 体积变化系数 (RDW)、和平均胞内血红蛋白质量 (MCH) 在全血液计数 (CBC) 中常规测量 (1)。最近,已经变得可能的是,鉴定并表征非常早期的 ( 数小时或几天龄 ) 循环 RBC( 网织红血球 )(2)。RBC 在自骨髓释放后在很少几天中经历体积和血红蛋白的快速降低 (3)。这种快相继之以长得多的较慢降低期 (4-7),期间体积和血红蛋白受到共调节 (8) ;见图 1A。

[0054] 网织红血球和所有循环 RBC 的概率分布的比较 ( 图 1A) 显示了体积和血红蛋白含量之间的关联随着细胞成熟,从网织红血球群体中约 0.40 的初始关联系数增加到全群体中的约 0.85。如此,虽然牵涉的许多分子机制是未知的,但是清楚的是平均 RBC 以如下的方式成熟,使得其血红蛋白浓度趋向于群体平均红细胞血红蛋白浓度 (MCHC),其在图 1A 和 1B 中以等浓度线显示。此共调节的结果是血红蛋白浓度的变化低于体积和血红蛋白含量的 (8)。

[0055] 本文中公开的参数显示了 TT 和 IDA 患者中的 RBC 在绝对和相对项两者中比它们会在正常条件下以小得多的体积和低得多的血红蛋白含量在外周中保持。其持久性可以由于这些贫血的不太有效的红细胞生成而反映清除的补偿性延迟。如此,必须存在可以改变 RBC 清除触发物的行为的机制。比较 TT 和 IDA 中的 RBC 清除与健康个体或 ACD 患者中的 RBC 清除可以提供鉴定触发物的新路径。对于健康个体,$v_c$ 的变化比 $\overline{v}$ 的变化小得多 (14),提示了触发物与 MCHC 线上的位置高度相关。

[0056] 贫血

[0057] 贫血是一种存在于血流中的血红蛋白量低于正常的状况。基于红细胞大小 ( 以 MCV,或均值红血球体积测量 ),贫血可以细分成三个大组。在红细胞具有正常大小 ( 即,80-100fL) 时诊断出正常红细胞性贫血。在红细胞大小大于正常,即大于 100fL 的 MCV 时诊断出大红细胞性贫血 ( 通常由 B12 缺乏引起 )。小红细胞性贫血是在红细胞小于正常,即 MCV<80fL 时。

[0058] 诊断前贫血 / 预测贫血的形成

[0059] 本文中呈现的模型和方法通过寻找清除延迟的体征来提供在明显的临床贫血形成前鉴定隐性或代偿性贫血患者的潜在方式。在具有正常的 CBC,至少 30 且不超过 90 天后

接着有另一正常 CBC 或临床 IDA 的独立的一组患者中测试此可能性。对于每个患者样品，将所有细胞的 (v,h) 坐标投影到 MCHC 线上，并且沿着此线在均值的 85%($P_{0.85}$) 下为概率密度积分。图 3A-C 显示了一名患者的 CBC 从正常演变成隐性 IDA 及最终 IDA。中间小图（图 3B) 中显示的 CBC 是临床上不显著的，但是 $P_{0.85}$ 是异常的，预测没有出现并且达到医学注意 51 天的贫血。图 3D 显示了来自保持健康的患者的 20 个正常 CBC 和来自 30-90 天后形成 IDA 的患者的 20 个正常 CBC 的 $P_{0.85}$ 值。$P_{0.85}$ 值以 75% 的灵敏度和 100% 特异性预测 IDA。参见实施例 3,“预测铁缺乏贫血”。目前的护理标准在此群体中具有 0% 的灵敏度，因为所有 CBC 是“正常的”。与统计学回归方法（其经常依赖于合并多次测量和来自不同来源和不同时间点的信息类型）形成对比，此基于模型的预测仅依赖于一个时间点时的单次 CBC 测量 (15)。

[0060] 如此，可以使用所描述的方法来诊断前贫血，或者在其形成前至少 30-90 天预测受试者中贫血的形成，例如，在受试者仍然具有其它方面正常的 CBC 时（例如，其中受试者的 HCT, MCV, RDW, MCH, MCHC, RBC 和 HGB 水平均在正常范围内）。

[0061] 正常的 CBC 值可以随实施分析的实验室而变化，但是一般，正常的数值如下。

[0062]

| 参数 | 数值 |
|---|---|
| 血细胞比容(HCT)<br>(随高度而变化) | 男性：40.7至50.3%。<br>女性：36.1至44.3% |
| 平均红细胞体积(MCV) | 80至95飞升(femtoliter) |
| 红细胞分布宽度(RDW) | 11.5 - 14.5% |
| 平均细胞血红蛋白(MCH) | 27至31 pg/细胞 |
| (MCHC) | 32至36 gm/dL |
| 红细胞(RBC)计数<br>(随高度而变化) | 男性：470至610万个细胞/uL<br>女性：420至540万个细胞/uL |
| 血红蛋白(HGB)<br>(随高度而变化) | 男性：13.8至17.2 gm/dL<br>女性：12.1至15.1 gm/dL |

[0063] 所述方法包括在来自受试者的样品中测定每个细胞的细胞体积和 HGB 含量（例如，HGB 质量或浓度），计算样品中所有细胞间体积和 HGB 含量值的分布，计算每个细胞体积和血红蛋白的转化，例如，如图 5 中显示的，计算样品中所有细胞的此类转化的分布，并测定其转化低于转化体积和 HGB 含量的阈值水平的细胞的百分比，例如，如图 3A-D 和 8A-D 中显示的。在一些实施方案中，所述方法包括例如在包含来自受试者的红细胞的样品中，测定群体平均红细胞血红蛋白浓度 (MCHC)；计算样品中所有细胞间体积和 HGB 含量值的分布，计算每个细胞的体积和血红蛋白的转化，例如，如图 5 中显示的，计算样品中所有细胞的此类转化的分布，测定样品中红细胞的分数以提供样品分数，所述样品的转化降到均值转化的阈值百分比下；并比较样品分数与参照分数，其中低于参照分数的样品分数的存在指示受试者有形成贫血，例如 IDA 的风险。阈值百分比和参照水平的数值可以使用本领域中已知的方法来选择，并且可以代表使灵敏度和特异性最大化的数值。在一些实施方案中，阈值百分比是群体均值 MCHC 的 70%, 75%, 80%, 85%, 90% 或 95%。在一些实施方案中，参照分数是

约 0.10,0.11,0.12,0.13,0.14 或 0.15。本领域技术人员会容易地能够使用已知的统计学方法来鉴定最佳的阈值百分比和参照分数。

[0064] 差异诊断 :珠蛋白生成障碍性贫血对铁缺乏贫血

[0065] 小红细胞性贫血的差异诊断包括铁缺乏贫血 (IDA)、珠蛋白生成障碍性贫血性状 (TT)、慢性病贫血 (ACD)、和其它贫血原因。

[0066] ACD 通常牵涉血细胞比容降低至低于正常的下限的不超过 20%、正常或高铁蛋白,以及低或正常总铁结合能力。若降低的 RBC 清除阈值 ($v_c$) 代表补偿贫血的适应性生理应答,则可能会预期本文对 ACD 看到的正常 $v_c$,其中贫血自身可以代表适应性生理应答 (18)。患有慢性病的受试者中贫血的存在通常提示 ACD 的诊断。通常与 ACD 有关的慢性病包括自身免疫性病症,例如,克罗恩 (Crohn) 氏病、系统性红斑狼疮、类风湿性关节炎、和溃疡性结肠炎 ;癌症,例如淋巴瘤和何杰金 (Hodgkin) 氏病 ;慢性肾病 ;肝病,例如硬化 ;和慢性感染,例如细菌性心内膜炎、骨髓炎 ( 骨感染 )、HIV/AIDS、乙肝或丙肝。参见例如 Gardner 和 Benz Jr., “Anemia of chronic diseases.” 于 :Hoffman 等编 Hematology:Basic Principles and Practice. 第 5 版 Philadelphia,Pa:Elsevier Churchill Livingstone;2008:chap37。

[0067] IDA 通常牵涉血细胞比容降低、低 MCV、和低铁蛋白。轻度 IDA 通常牵涉血细胞比容降低至低于正常的下限的不超过 20%。IDA 患者可以显示正常 MCV 及正常血细胞比容的历史证据。

[0068] TT 通常与高血红蛋白 A2 分数,或者与一处或多处 alpha 珠蛋白基因突变的存在有关。TT 经常牵涉血细胞比容通常降低至低于正常的下限的不超过 20%、低 MCV、和正常铁蛋白。TT 是世界上最常筛选的条件之一,但是现有的诊断方法是非常昂贵的或者具有低得不可接受的诊断准确度,假阳性率多至 30%(19-20)。本文中所描述的模型和方法提供了区别 IDA 和 TT 及可能其它贫血原因的新的且可能更为准确的方式。

[0069] 图 2 显示了 $D_h$ 区别 IDA 和 TT,即两种最常见的小红细胞性贫血原因。通过分析 10 份训练样品来建立例示性 $D_h$ 阈值数值 0.0045。然后,分析 50 份独立的患者样品,其中可以确信地建立轻度 IDA 或 TT 的诊断,并且计算 $D_h$。图 4 显示了此 $D_h$ 阈值具有 98% 的诊断准确性,正确鉴定 22/22IDA 病例和 27/28TT 病例,而且工作性能比其它发表的方法好 6-41%(20)。参见实施例 4,“区别诊断小红细胞性贫血”。

[0070] 如此,本文中所描述的方法可以包括通过如下的方法在患有小红细胞性贫血的受试者中在铁缺乏贫血 (IDA) 和珠蛋白生成障碍性贫血性状 (TT) 及可能其它贫血原因之间做出区别诊断,所述方法包括在来自受试者的样品中测定与平均血红蛋白含量 ($D_h$) 的波动幅度数值,并比较 $D_h$ 与参照数值。高于参照数值的 $D_h$ 的存在指示受试者患有 IDA,而低于参照数值的 $D_h$ 的存在指示受试者患有 TT。

[0071] 优化红细胞生成刺激剂 ( 例如,红细胞生成素 (EPO)) 和 / 或铁补充疗法

[0072] 也可以使用本文中所描述的方法来优化 EPO 或铁补充疗法。例如,所述方法可以包括在来自经历 EPO 治疗的受试者的样品中测定标准化临界体积 $v_c$。

[0073] 在一些实施方案中,调节 EPO 或铁补充的剂量,从而以期望的水平,例如高于或低于阈值水平的水平,或在期望范围内的水平维持 $v_c$ 或 $D_h$。例如,如本文中记录的,正常健康个体中的 $v_c$ 是群体均值体积 $\bar{v}$ 的约 80%,或约 72fL( 典型的 MCV 为 90fL)。如此,在一些实

施方案中，所述方法包括调节EPO或铁补充剂量以维持受试者中$\overline{v}$的大于70%的$v_c$；受试者中$\overline{v}$的大于75%的$v_c$；受试者中$\overline{v}$的75-85%的$v_c$。在一些实施方案中，所述方法包括调节治疗以维持大于70%的$v_c$；或70-75%的$v_c$。在一些实施方案中，随时间监测$v_c$或$D_h$，并且调节对受试者施用的EPO或铁补充的剂量以保持高于或低于选择的阈值，或在给定的范围内的$v_c$或$D_h$。

[0074] 鉴定具有升高的GI病症风险的受试者

[0075] IDA或其它方面不清楚的贫血经常是严重状况，包括胃肠病症诸如结肠癌(16)和儿童营养不良(17)的初始表现。其它与IDA有关的GI病症包括结肠直肠(例如，结肠)癌；胃肠道溃疡(胃、消化性，盲肠)；憩室炎；缺血性肠(ischemic bowel)；胃癌；胃炎；食管炎；GI息肉；炎性肠病(克罗恩氏病、溃疡性结肠炎)；和乳糜泻。藉由此方法早期检测或预测贫血会实现对此类疾患的更快响应。如此，可以使用本文中的方法来进行筛选或推荐额外筛选，例如针对GI病症(例如，使用结肠镜检查术进行)筛选或营养评估。然后，可以更快启动合适的患者病情检查(work-up)，例如，实施结肠镜检查术，或者，然后，可以更快启动合适的治疗，例如，规定铁补充。

[0076] 检测血掺

[0077] 运动员的表现(特别在需氧运动中)可以受到存在的RBC数目影响；RBC越多，将氧从肺转运并递送到工作肌肉的能力越大。可以依靠输血(称为血掺)，或使用红细胞生成刺激物药剂诸如红细胞生成素、类似物和模拟物获得红细胞的急剧且短暂增加。众人皆知地，血掺(doping)可以难以检测。参见例如Jelkmann和Lundby,Blood.2011年9月1日;118(9):2395-404;Segura等，“Current strategic approaches for the detection of blood doping practices,”Forensic Sci Int.2011年8月31日[印刷前电子出版]。可以使用本方法来检测RBC体积和HGB含量的异常分布的存在，这会指示可能使用血掺或红细胞生成刺激物药剂。例如，在一些实施方案中，在来自怀疑血掺或使用红细胞生成刺激物药剂的受试者的样品中测定与平均血红蛋白含量($D_h$)的波动幅度和与平均细胞体积($D_v$)的波动幅度中一项或两项的数值，并且高于参照数值的$D_h$和/或$D_v$的存在指示存在或使用血掺或者使用红细胞生成刺激物药剂。

[0078] 样品、系统、软件、和测定方法

[0079] 使用外周血样品实施本文中所描述的方法，所述外周血样品使用使RBC保持完整的已知收集方法(例如，用合适量的真空(抽吸)和大得足以容许RBC在没有实质性溶血的情况下收集的针，例如至少25g或更大的针抽取血液)获得。优选地，在收集的24、12、或6小时内进行测量。可以使用可以测量RBC体积(例如，使用低角度(2°-3°)散射检测)和血红蛋白质量或浓度(例如，使用高角度(5°-15°)散射检测)两者的本领域已知的任何方法或装置进行网织红血球和CBC测量。例示性方法记载于US20110178716,US20110164803,US20110149061,20110077871,20110070606和20110070210。

[0080] 在一些实施方案中，使用血液分析仪，例如，手动、半自动化、或自动化血液学分析仪(其例子是本领域已知的，并且记载于例如美国专利No.5,017,497,5,266,269,5,378,633,5,631,165,5,812,419,6,228,652,6,524,858,6,320,656,7,324,194和7,981,681，及公布的美国专利申请US20080153170,US20080158561,US20080268494,US20110178716,20110077871和20110070606，其公开内容通过提及完整并入本文)进行测

量。可用于本方法的血液分析仪可以使用本领域中已知的任何检测方法，例如流式细胞术或基于光学或图像的分析或基于阻抗的。可用于本方法的血液分析仪通常会是那些能够测量所有 CBC 参数的。具体地，分析仪应当能够至少测定红细胞 (RBC) 细胞体积 (CV)，和细胞血红蛋白浓度 (CHC) 或细胞血红蛋白质量 (CH)。

[0081] 许多血液学分析仪型号是商品化的，例如来自 Abbott Laboratories(Abbott Park, IL, United States)(例如，Cell-Dyn Sapphire)；和 Siemens(Deerfield, IL, United States)(例如，Advia120 或 2120 自动化血液分析仪)。其它制造商包括 Beckman Coulter, Inc.(Fullerton, CA, United States)；TOA Medical Electronics Co.,(Kobe, Japan)；Constitution Medical(Boston, MA)；和 HORIBA ABX Inc(Irvine, CA, United States)。

[0082] 本发明还提供了血液学分析仪系统，其包含用于测量临床样品的检测模块；和计算装置，其与所述检测模块通信，并且包含基于检测模块的输出来检测下列一项或多项的编程：(i) 群体平均红细胞血红蛋白浓度 (MCHC) 和样品中红细胞的分数以测定样品分数，所述样品的转化体积和血红蛋白含量降到所有转化体积和血红蛋白含量平均值的阈值百分比下；(ii) 群体中细胞间血红蛋白含量降低 ($D_h$) 速率的变化幅度；(iii) 群体中细胞间细胞体积降低 ($D_v$) 速率的变化幅度；和 / 或 (iv) 标准化临界体积 ($v_c$) 或其它清除阈值。在一些实施方案中，计算装置是一种使用来自血液分析仪的输入的分开的计算机。在一些实施方案中，计算装置整合入血液分析仪装置中，或者是血液分析仪装置的一部分。

[0083] 一般地，检测模块包含分析室，该分析室配置为容纳包含红细胞的样品，例如全血，以进行分析。在一些实施方案中，检测模块包含配置为分析样品的流式细胞仪。在一些实施方案中，检测模块包含配置为分析样品的光学或图像分析仪。

[0084] 本文中还提供了包含计算下列一项或多项的编程的计算机可读介质：(i) 群体平均红细胞血红蛋白浓度 (MCHC) 和样品中红细胞的分数以测定样品分数，所述样品的转化体积和血红蛋白含量降到所有转化体积和血红蛋白含量平均值的阈值百分比下；(ii) 群体中细胞间血红蛋白含量降低 ($D_h$) 速率的变化幅度；(iii) 群体中细胞间细胞体积降低 ($D_v$) 速率的变化幅度；和 / 或 (iv) 标准化临界体积 ($v_c$) 或其它清除阈值。

[0085] 在一些实施方案中，来自血液学分析仪的数据由计算装置接收 (例如，红细胞 (RBC) 细胞体积 (CV)、平均细胞体积 (MCV)、细胞血红蛋白浓度 (CHC) 或细胞血红蛋白含量 (CH)、平均细胞血红蛋白浓度 (MCHC)、和平均细胞血红蛋白含量 (MCH)、及其群体统计学)，所述计算装置然后执行含有用于计算下列一项或多项的如本文中所描述的算法的编程：(i) 群体平均红细胞血红蛋白浓度 (MCHC) 和样品中红细胞的分数以测定样品分数，所述样品的转化体积和血红蛋白含量降到所有转化体积和血红蛋白含量平均值的阈值百分比下；(ii) 群体中细胞间血红蛋白含量降低 ($D_h$) 速率的变化幅度；(iii) 群体中细胞间细胞体积降低 ($D_v$) 速率的变化幅度；和 / 或 (iv) 标准化临界体积 ($v_c$) 或其它清除阈值。

[0086] 在一些实施方案中，计算装置是一种使用来自血液分析仪的输入的分开的计算机。在一些实施方案中，计算装置整合入血液分析仪装置中，或者是血液分析仪装置的一部分。

[0087] 编程可以在物理存储或传输媒体中提供。然后，接收指令的计算装置 (例如，分开的装置或作为分析仪系统的一部分的信息加工模块) 可以执行算法和 / 或自本方法获得的

过程数据。计算机可读的存储介质的例子包括软盘、磁带、CD-ROM、硬磁盘驱动器、ROM 或集成电路、磁光盘、或计算机可读卡诸如 PCMCIA 卡等，无论此类装置是对于计算机而言是内部的或外部的。含有信息的文件可以“存储”于计算机可读介质上，其中“存储”意指记录信息，使得它在后来的日期由本地或远程网络的计算机可存取并可检索。在一些实施方案中，每次运行样品自动执行本文中所描述的方法。

[0088] 图 14 显示了可以与本文所描述的技术一起使用的计算机装置 700 和移动式计算机装置 750 的例子。计算装置 700 意图代表各种形式的数字计算机，包括例如膝上型、台式、工作站、个人数字助理、服务器、刀片式服务器、主机、和其它合适的计算机。计算装置 750 意图代表各种形式的移动装置，包括例如个人数字助理、移动电话、智能手机、及其它类似的计算装置。本文显示的组件、其连接和联系、及其功能意图仅为例子，而并不意图限制此文件中所描述和 / 或要求保护的技术的执行。

[0089] 计算装置 700 意图代表各种形式的数字计算机，包括例如膝上型、台式、工作站、个人数字助理、服务器、刀片式服务器、主机、和其它合适的计算机，包括整合到分析仪系统或装置中的计算机。计算装置 750 意图代表各种形式的移动装置，包括例如个人数字助理、移动电话、智能手机、及其它类似的计算装置。本文显示的组件、其连接和联系、及其功能意图仅为例子，而并不意图限制此文件中所描述和 / 或要求保护的技术的执行。

[0090] 计算装置 700 包含处理器 702、储存器 704、存储装置 706、与储存器 704 和高速扩充端口 710 连接的高速用户界面 708、和与低速总线 714 和存储装置 706 的低速用户界面 712。组件 702、704、706、708、710、和 712 中的每个使用多个总线相互连接，并且可以在共同的母板上或适当时以其它方式安装。处理器 702 可以处理计算装置 700 内执行的指令，包括存储器 704 中或存储装置 706 上存储的指令以在外部输入 / 输出装置，包括例如与高速用户界面 708 偶联的显示器 716 上显示关于 GUI 的图形信息。在其它实现中，在适当时，可以使用多个处理器和 / 或多个总线，以及多个存储器和存储器类型。还有，可以连接多个计算装置 700，每个装置提供必要的操作的部分（例如，作为服务器存储体 (server bank)、一组刀片式服务器、或多处理器系统）。

[0091] 存储器 704 存储计算装置 700 内的信息。在一个实现中，存储器 704 是一个或多个易失性存储器单元。在另一个实现中，存储器 704 是一个或多个非易失性存储器单元。存储器 704 也可以是另一种形式的计算机可读的介质，包括例如磁盘或光盘。

[0092] 存储装置 706 能够为计算装置 700 提供海量存储器。在一个实现中，存储装置 706 可以是或者含有计算机可读的介质，包括例如软盘装置、硬盘装置、光盘装置、或磁带装置、闪速存储器或其它类似的固态存储器装置、或装置阵列，包括存储区网络或其它配置中的装置。计算机程序产品可以在信息载体中有形体现。计算机程序产品也可以含有如下的指令，其在执行时实施一种或多种方法，包括例如上文描述的那些方法。信息载体是计算机或机器可读的介质，包括例如存储器 704、存储装置 706、处理器 702 上的存储器，等等。

[0093] 高速控制器 708 管理计算装置 700 的带宽密集操作，而低速控制器 712 管理较低的带宽密集操作。此类功能分配仅是一个例子。在一个实现中，高速控制器 708 与存储器 704、显示器 716（例如，藉由图形学处理器或加速器），以及与高速扩充端口 710（其可以接受各种扩充卡（未显示））偶联。在一个实现中，低速控制器 712 与存储装置 706 和低速扩充端口 714 偶联。低速扩充端口（其可以包括多个通信端口（例如，USB、Bluetooth®、以

太网、无线以太网））可以例如藉由网络适配器与一个或多个输入／输出装置，包括例如键盘、打印装置、扫描仪，或连网装置，包括例如开关或路由器偶联。

[0094] 计算装置 700 可以以许多不同形式实现，如图中所显示的。例如，它可以以标准服务器 720，或者在一组此类服务器中多次实现。它也可以作为架式服务器系统 724 的一部分实现。另外或作为备选，它可以在个人计算机，包括例如膝上型电脑 722 中实现。在一些例子中，来自计算装置 700 的组件可以与移动装置（未显示），包括例如装置 750 中的其它组件组合。每个此类装置可以含有一个或多个计算装置 700、750，并且整个系统可以由彼此通信的多个计算装置 700、750 构成。

[0095] 计算装置 750 包括处理器 752、存储器 764、输入／输出装置，包括例如显示器 754，通信用户界面 766、和收发器 768，等等。也可以给装置 750 提供存储装置，包括例如微驱动器 (microdrive) 或其它装置，以提供额外的存储。组件 750, 752, 764, 754, 766 和 768 中的每个使用多个总线相互连接，并且几个组件可以在共同的母板上或适当时以其它方式安装。

[0096] 处理器 752 可以执行计算装置 750 内的指令，包括存储器 764 中存储的指令。处理器可以作为包含分开的且多个模拟和数字处理器的芯片的芯片组实现。处理器可以提供例如装置 750 的其它组件的协调，包括例如控制用户界面、由装置 750 运行的应用、和装置 750 的无线通信。

[0097] 处理器 752 可以藉由与显示器 754 偶联的显示器用户界面 756 和控制器用户界面 758 与用户通信。显示器 754 可以例如是 TFT LCD（薄膜晶体管液晶显示器）或 OLED（有机发光二极管）显示器，或其它合适的显示器技术。显示器用户界面 756 可以包含适合于驱动显示器 754 以对用户呈现图形和其它信息的电路。控制用户界面 758 可以自用户接收命令，并且转化它们以提交给处理器 752。另外，外部用户界面 762 可以与处理器 742 通信，从而实现装置 750 与其它装置的近区通信。外部用户界面 762 可以例如提供有线通信（在一些实现中，或者无线通信（在其它实现中），并且也可以使用多个用户界面。

[0098] 存储器 764 存储计算装置 750 内的信息。存储器 764 可以作为一个或多个计算机可读的介质、易失性存储器单元、或非易失性存储器单元实现。也可以提供扩充存储器 774，并藉由扩充用户界面 772 与装置 750 连接，所述扩充用户界面 772 可以包括例如 SIMM（单直插式存储器模块 (Single In Line Memory Module)）卡用户界面。此类扩充存储器 774 可以为装置 750 提供额外的存储空间，或者也可以为装置 750 存储应用或其它信息。具体地，扩充存储器 774 可以包括实施或补充上文所描述的过程的指令，并且可以还包括安全信息。如此，例如，扩充存储器 774 可以以装置 750 的安全性模块提供，并且可以用容许装置 750 的安全使用的指令编程。另外，可以与别的信息一起藉由 SIMM 卡提供安全应用，包括例如以不可破解 (non-hackable) 的方式在 SIMM 卡上放置鉴定信息。

[0099] 存储器可以包括例如闪速存储器和／或 NVRAM 存储器，如下文所讨论的。在一个实现中，计算机程序产品在信息载体中有形体现。计算机程序产品含有在执行时实施一种或多种方法，包括例如上文所描述的那些方法的指令。信息载体是计算机或机器可读的介质，包括例如存储器 764、扩充存储器 774、和／或处理器 752 上的存储器，其可以例如在收发器 768 或外部用户界面 762 上接收。

[0100] 装置 750 可以藉由通信用户界面 766 无线通信，所述通信用户界面 766 在必需的

情况中可以包含数字信号处理电路。通信用户界面 766 可以提供各种模式或方案下的通信，包括例如 GSM 语音呼叫、SMS、EMS、或 MMS 电报、CDMA、TDMA、PDC、WCDMA、CDMA2000、或 GPRS 等。可以例如藉由射频收发器 768 发生此类通信。另外，可以发生短程通信，包括例如使用 Bluetooth®、WiFi、或其它此类收发器（未显示）进行。另外，GPS（全球定位系统）接收器模块 770 可以将别的导航和位置相关无线数据提供给装置 750，其可以在适当时由装置 750 上的应用运行使用。

[0101] 装置 750 也可以使用音频编解码器 760 在听觉上通信，所述音频编解码器 760 可以自用户接收口头信息，并将其转化成可用的数字信息。同样地，音频编解码器 760 可以为用户产生可听声，包括例如，藉由扬声器，例如在装置 750 的听筒中。此类声音可以包括来自声音电话呼叫的声音，可以包括记录的声音（例如，语音信息、音乐文件等），而且还可以包括由装置 750 上的应用操作产生的声音。

[0102] 计算装置 750 可以以许多不同形式实现，如图中显示的。例如，它可以作为移动电话 780 实现。它也可以作为智能手机 782、个人数字助理、或其它类似的移动装置的一部分实现。

[0103] 可以在数字电子电路、集成电路、空间设计 ASIC（专用集成电路 (application specific integrated circuits)）、计算机硬件、固件、软件、和 / 或其组合中实现本文所描述的系统和技术的各种实现。这些各种实现可以包括在可编程系统上可执行和 / 或可解读的一个或多个计算机程序中的执行，所述可编程系统包含至少一个可编程的处理器（其可以是特殊或一般目的）、至少一个输入装置、和至少一个输出装置，所述可编程的处理器相偶联以自存储系统接受数据和指令，并将信息和指令与存储系统通信。

[0104] 这些计算机程序（又称为程序、软件、软件应用或代码）包含可编程处理器的机器指令，并且可以以高水平程序和 / 或面向对象的编程语言，和 / 或以汇编 / 机器语言实现。如本文中所使用的，术语机器可读的介质和计算机可读的介质指用于对可编程处理器提供机器指令和 / 或数据的计算机程序产品、设备和 / 或装置（例如磁盘、光盘、存储器、可编程逻辑装置 (PLD)），包括接受机器指令的机器可读介质。

[0105] 为了提供与用户的交互，可以在计算机上实现本文所描述的系统和技术，所述计算机具有用于对用户显示信息的显示器装置（例如，CRT（阴极射线管）或 LCD（液晶显示器）监视器）和用户可以将输入提供给计算机的键盘和打印装置（例如，鼠标或跟踪球）。也可以使用其它种类的装置来提供与用户的交互；例如，对用户提供的反馈可以是感觉反馈（例如，视觉反馈、听觉反馈、或触觉反馈）形式；并且来自用户的输入可以以包括听觉、语音、或触觉输入的形式接收。

[0106] 可以在计算系统中实现本文所描述的系统和技术，所述计算系统包含后端组件（例如，作为数据库服务器），或包含中间件组件（例如，应用服务器），或包含前端组件（例如，具有图形用户界面或网络浏览器的客户计算机，藉由该客户计算机，用户可以与本文所描述的系统和技术的实现交互），或所述后端、中间件、或前端组件的组合。系统的组件可以通过数字数据通信的形式或介质（例如通信网络）相互连接。通信网络的例子包括局域网络 (LAN)、广域网络 (WAN)、和因特网。

[0107] 计算系统可以包含客户机和服务器。一般地，客户机和服务器彼此是远程的，并且通常藉由通信网络交互。依靠在相应的计算机上运行且彼此具有客户机 - 服务器关系的计

算机程序发生客户机和服务器的关系。

[0108] 在一些实现中，本文中所描述的机械 (engine) 可以是分开的、组合的或整合到单一或组合的机械中。图中所描绘的机械并不意图将本文所描述的系统限于图中所显示的软件构造。

## 实施例

[0109] 本发明在以下实施例中进一步描述，所述实施例不限制权利要求书中所描述的本发明范围。

[0110] 实施例 1 :模型开发

[0111] 体内个别 RBC 在其寿命过程期间的体积和血红蛋白调节是非常复杂且难以了解的。了解 RBC 大群体的平均行为可能是更易处理的。为了了解此群体水平行为，开发一种 RBC 成熟和清除模型，其描述 RBC 群体的动力学。该模型将随时间 (t) 的平均 RBC 的体积 (v) 和血红蛋白 (h) 动力学分解成确定性降低 (deterministic reduction)(f) 和随机波动 (random fluctuation)(ζ)，其具体函数形式可以有所变化，方程式 1 中显示了一个例子，其中 v 和 h 按其群体均值 $(\bar{v},\bar{h})$ 调整，而 t 按平均细胞龄 $(\bar{\tau})$ 调整。基于来自先前报告 (3-5,7) 的数据，将两项参数引入确定性分量 (deterministic component) 中 :快变化 (fast change)(β)，其效应在 RBC 接近 MCHC 线前占优势，和慢变化 (slow change)(α)。随机波动可以建模为具有均值 0 的高斯或类似分布的随机变量和由扩散张量 (diffusion tensor)2D 给予的方差，如图 1B 和方程式 1 中所显示的 (The random fluctuation can be modeled as a Gaussian or similarly distributed random variable with mean zero and variance given by a diffusion tensor 2D, as shown in FIG. 1B and Equation 1)。

[0112] $$\begin{bmatrix} \frac{dv}{dt} \\ \frac{dh}{dt} \end{bmatrix} = \mathbf{f} + \zeta \qquad (1)$$

[0113] $$\mathbf{f} = \begin{cases} \alpha \cdot e^{\beta_v (v-h)} \\ \alpha \cdot e^{\beta_h (h-v)} \end{cases}$$

[0114] $$\zeta = \begin{cases} N(0,2D_v) \\ N(0,2D_h) \end{cases}$$

[0115] 如同一般地逆问题及特别地人病理生理学一样 (9)，此问题在如下的意义上是不适定的 (ill-posed)，即 f 的不同函数形式会再现体内动力学。f 的精确函数形式不是必需的。此模型的性能依赖于快和慢确定性动力学及随机波动的定性组合。表 1A-B 显示了可以在所述模型中使用的不同类型的函数形式。

[0116] 表 1A :f 的函数形式

[0117]

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

[0118] 表 1B :d 的函数形式

[0119]

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

[0120] 图 9A-C 显示了与 f 的形式 A-C 对应的速度场，而图 10A-B 显示了清除函数 (d) 形式 A 和 B。 τ 是群体中细胞的均值龄。下文列出 Δ 的定义。

[0121] 对确定性演变 (evolution)(f) 和清除函数 (d) 的不同函数形式建立参数。定性结果对于这些不同函数形式是一致的，提示了这些结果代表体内病理生理学的特征，而不是数据的过度拟合 (overfitting)。表 1A-B 中显示了函数形式的详情，并且在图 2、11、和

12 的盒式图中显示了形式 A-C 的估值。

[0122] 在本模型中，典型的个别细胞的体积和血红蛋白含量的随机波动和确定性耗散或降低由朗之万 (Langevin) 方程描述，所述朗之万方程通常用于为势能中的布朗运动建模 (10)。然后，RBC 的整个循环群体的动力学可以由时间依赖性联合体积 - 血红蛋白概率分布的主方程 (P(v,h,t)) 描述，该主方程可以由 Fokker-Planck 方程 (10-11) 逼近。方程式 2 描述了此联合体积 - 血红蛋白分布的概率密度的漂移 (drift)(f)、扩散 (D)、出生 (birth, b)、和死亡 (d)。

[0123] $$\frac{\partial P}{\partial t} = -\nabla \cdot (P\mathbf{f}) + \nabla \cdot (\mathbf{D} \cdot \nabla P) + b(v,h,t) - d(v,h,t)P \qquad (2)$$

[0124] $$\mathbf{D} = \begin{bmatrix} D_v & 0 \\ 0 & D_h \end{bmatrix}$$

[0125] 出生和死亡过程造成不断对群体添加及自群体除去的 RBC。在健康和轻度疾病的状态中，添加的细胞总数等于消除的总数：$\iint d(v,h)P\,dv\,dh = \iint b(v,h)\,dv\,dh$。消除 RBC 的精确触发物和机制没有得到完全了解 (12)，但是经验测量诸如那些在图 1A 中显示的那些测量提示了沿着 MCHC 线存在着阈值 ($v_c$)，超出该阈值，已经消除大多数 RBC。

[0126] 基于经验 RBC 分布的观察结果，将 RBC 消除的概率建模为 RBC 体积和血红蛋白含量的函数。将体积 - 血红蛋白含量平面中的每个 RBC 位置投影到 MCHC 线上，并且将清除 (d) 概率不同地限定为例如在此线上从此投影点到阈值 $v_c$ 的距离的 s 形曲线 ( 图 10A) 或阶梯 ( 图 10B) 函数。参见图 1 和 5 及表 1B。方程式 3 量化此关系。

[0127] $$d(v,h) = \frac{1}{1+e^{\Delta}} \qquad (3)$$

[0128] 或

[0129] $$d(v,h) = \begin{Bmatrix} 1 & \Delta \le 0 \\ 0 & \Delta > 0 \end{Bmatrix}$$

[0130] $$\Delta(v,h) = 100 \cdot \frac{\cos(\theta)\sqrt{(v\bar{v})^2 + (h\bar{h})^2} - v_c\sqrt{\bar{h}^2 + \bar{v}^2}}{v_c\sqrt{\bar{h}^2 + \bar{v}^2}}$$

[0131] $$\theta = \tan^{-1}\left(\frac{\bar{h}}{\bar{v}}\right) - \tan^{-1}\left(\frac{h\bar{h}}{v\bar{v}}\right)$$

[0132] CBC 测量在个体间有所变化，但是对于健康个体不显著改变 (13)，指示这些动力学过程在体内达到稳态 $P_\infty$，其中 $\lim_{t\to\infty} P(v,h,t) \to P_\infty(v,h), i.e., \frac{\partial P_\infty}{\partial t} = 0$。

[0133] 对于给定的一组参数，可以以在病理生理学范围外的体积和血红蛋白含

量的消失概率边界条件和等于凭经验测量的网织红血球分布的初始条件使用一阶 $\left(\mathbf{J}=\frac{\Delta_k[f\cdot P](v)}{k}+\frac{\Delta_k[f\cdot P](h)}{k}\right)$ 和二阶 $\left(\mathbf{L}=D_v\frac{\delta_k^2[P](v)}{k^2}+D_h\frac{\delta_k^2[P](h)}{k^2}\right)$ 空间导数的有限差分逼近在数字上解析方程式 2。用恒定网孔宽度 (mesh width) 离散化体积 - 血红蛋白含量平面，并且其以等于每个网孔细胞中的概率密度的变量的向量 (P) 代表。这里报告的模拟结果用网孔宽度 1.8fL 沿着体积轴和 1.8pg 沿着血红蛋白含量轴执行。此网孔宽度与经验体积和血红蛋白含量测量的分析分辨率相当。使用较小的网孔宽度 (1.2fL 和 1.2pg) 确认所有结果。使用空间导数的逆风有限差逼近 (upwind finite difference approximation) 为对流分布 (f) 数字建模。使用 MATLAB ode15s 积分器为普通微分方程的所得线性系统积分，并迭代，直至达到稳态 (P ∞ )。

[0134] 还分析测定数字问题 (numerical problem)(P ∞ ) 的稳态分布 ;演变 (J+L) 和消除 (d) 项的数字逼近是线性算子，并且调整出生过程的整数是等于均值龄两倍的倒数 $\left(\frac{1}{2\bar{\tau}}\right)$ 的常数。可以转化线性算子，产生以均值细胞龄为指数的稳态分布家族，如方程式 4 中显示的。

[0135] $$\frac{d\mathbf{P}_\infty}{dt}=0=-\mathbf{J}\cdot\mathbf{P}_\infty+\mathbf{L}\cdot\mathbf{P}_\infty+d\cdot\mathbf{P}_\infty+\mathbf{P_0}\int_h\int_v d(v,h)=$$

[0136] $$(-\mathbf{J}+\mathbf{L}+d)\cdot\mathbf{P}_\infty+\mathbf{P_0}\frac{1}{2\bar{\tau}}\Rightarrow$$

[0137] $$\mathbf{P}_\infty=-(-\mathbf{J}+\mathbf{L}+d)^{-1}\mathbf{P_0}\frac{1}{2\bar{\tau}} \qquad (4)$$

[0138] 在通过分析和有限差方法获得的稳态分布之间可忽略差异。

[0139] 通过适当选择参数 ( α 、β 、D、和 $v_c$)，本文中所描述的模型如实再现健康个体中 RBC 群体的观察到的分布。

[0140] 实施例 2 :确认

[0141] 为了测试模型是否可以区别健康个体中 RBC 群体的动力学与贫血个体的，对患有具有不同根本病因学的三种常见贫血形式 :慢性病贫血 (ACD)，即一种炎性状况 ;珠蛋白生成障碍性贫血性状 (TT)，即一种遗传病症 ;和铁缺乏贫血 (IDA)，即一种营养状况的个体获得 CBC 和网织红血球测量 (14)。选择每种贫血的轻度病例，其中 RBC 群体特征似乎是稳定的，并且类稳态假设是合理的，几个明显健康的对照亦然。

[0142] 在得到合作健康护理机构审查委员会 (Partners Healthcare Institutional Review Board) 批准的研究方案下从三级护理成人医院的临床实验室获得血液样品和 CBC 结果。在 Siemens Advia2120 自动化分析仪上在收集 (21) 的 6 小时内进行网织红血球和 CBC 测量。

[0143] IDA 定义为血细胞比容轻微降低至低于正常的下限的不超过 20%、低 MCV、低铁

蛋白，以及正常 MCV 及正常血细胞比容的历史证据。排除急性疾病、急性出血 (acute bleeding)、在先前的 6 个月中输血、并发住院 (concurrent hospitalization)、慢性炎性疾病、或血红蛋白病患者。

[0144] TT 定义为高血红蛋白 A2 分数、或在存在 alpha 珠蛋白基因突变方面杂合性，以及血细胞比容降低到低于正常下限的不超过 20%、低 MCV、和正常铁蛋白。排除急性疾病、急性出血、在先前的 6 个月中输血、并发住院、慢性炎性疾病、或别的血红蛋白病患者。

[0145] ACD 定义为血细胞比容降低到正常下限下不超过 20%、正常或高铁蛋白、和低或正常总铁结合能力。排除急性疾病、急性出血、在先前的 6 个月中输血、并发住院、慢性炎性疾病、或血红蛋白病患者。

[0146] 对于每个患者样品，鉴定再现对所述患者观察到的稳态的最佳参数集（α、β、D 和 $v_c$）。使用模拟的稳态分布和测量的 CBC 分布之间的最小平方拟合来鉴定最佳拟合。在重复测试可用于相同个体的情况中，发现了通过 CBC 测量的分析变化解释拟合参数的任何变化。

[0147] 使用梯度和非梯度优化方法对每名患者鉴定参数空间的最佳邻域。使用患者的凭经验测量的网织红血球分布和初始随机选择的参数集作为起始点。使用方程式 4 计算所得的稳态 RBC 分布。然后，比较此计算的分布 ($P_\infty$) 与经验分布 (PCBC)。如下量化参数估值的质量，即计算等于离散化分布的标准化残数平方和的目标函数，如方程式 5 中显示的，其中 i 和 j 代表离散的体积 - 血红蛋白平面中的细胞指数。

[0148] $$C(P_{CBC}^{i,j}, P_{\infty}^{i,j}) = \sum_{i,j} \frac{\left(P_{CBC}^{i,j} - P_{\infty}^{i,j}\right)^2}{P_{CBC}^{i,j}} \qquad (5)$$

[0149] 一般地，目标函数提供两种分布间不相似（或相似性）程度的测量。可以基于目标函数的数值是否满足阈值条件来调节参数数值。例如，可以调节参数数值，直至目标函数的数值高于或低于阈值。若目标函数代表不相似的测量（例如平均平方差、绝对差的和，等等），则寻求通过调节参数降低目标函数。或者，若目标函数代表相似性的测量（例如相关系数、交互信息等），则寻求通过调节参数提高目标函数。在此实施例中，然后选择新的参数数值来降低此目标函数。使用 MATLAB 中的基于梯度的 (lsqnonlin 函数）和基于非梯度的 (fminsearch 和 patternsearch 函数）优化算法来搜索最佳参数。将所有参数约束为非负，并限定一致的初始参数空间以排除大于 1000 或小于 5 天的均值细胞龄。然后，使用拉丁超方 (latin hyper-square) 取样从此空间挑出初始参数。强加一个优化约束，将 α 限制为小得足以使均值细胞龄会大于 5 天且大得足以使均值细胞龄会小于 1000 天。图 6 显示了该模型为此健康患者提供可信的 PCBC 再现，其通过比较模拟的和测量的稳态概率分布进行。在沿着 MCHC 线投影时，此患者的经验分布具有接近该模型的略高密度以及在该模型的任一侧的区域中略低的密度。

[0150] 图 7 显示了从单一患者的超过 200 次优化获得的参数的最小局部最小值。一些结果具有高于轴范围的局部最小值。所有模拟间的最佳拟合为所有参数形成数值的小邻域，表明参数评估过程达到对于此患者的血液样品而言的定义明确的最佳邻域。

[0151] 图 2 中显示了健康和贫血个体的拟合参数，并在表 2 中列出了均值拟合参数。

[0152] 表 2：非维量（dimensional）和维量拟合参数的中值数值（在适当时）。

[0153]

| | 正常的 | ACD | TT | IDA |
|---|---|---|---|---|
| $\beta_v$ | 26 | 27 | 14 | 15 |
| $\beta_h$ | 16 | 15 | 5 | 12 |
| $\alpha$ | 0.05 (0.09 fL/d和 0.03 pg/d) | 0.05 (0.09 fL/d和 0.03 pg/d) | 0.13 (0.20 fL/d和 0.07 pg/d) | 0.13 (0.20 fL/d 和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) | $2.7 \times 10^{-5}$ ($4.9 \times 10^{-4}$ $pg^2$/d) | $2.7 \times 10^{-15}$ ($3.6 \times 10^{-14}$ $pg^2$/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) |

[0154] 对健康个体和那些患有贫血的个体衍生的最佳拟合参数间存在清楚的差异，并且不同贫血条件具有不同特征性参数集。例如，健康个体和 ACD 患者具有高 $\beta_v$ 和 $\beta_h$ 以及低 $\alpha$，即他们在快相期间比他们在慢相期间丧失相对更多的其体积和血红蛋白。比较而言，患有 TT 和 IDA 的患者在慢相期间比在快相中丧失相对更多的体积和血红蛋白。相对于健康个体，患有 ACD 的患者显示略微升高的 $D_v$ 和略微降低的 $D_h$，而 TT 与 $D_v$ 的更大升高以及实质性降低的 $D_h$ 有关。IDA 患者与健康个体具有相似的 $D_v$，并且在大多数个体的情况下显示比正常高超过 10 倍的 $D_h$ 显著升高。健康个体和那些患有 ACD 的个体中的标准化临界体积 $v_c$ 是 $\bar{v}$ 的约 80%，或约 72fL。大多数患有 TT 或 IDA 的患者通常具有降低的 $\bar{v}$ 和降低的 $\bar{h}$。图 2 显示了在 $\bar{v}$ 和 $\bar{h}$ 的绝对降低外，这些患者的 $v_c$ 进一步降低，而且在不同个体间显示大得多的变化性。

[0155] 实施例 3：预测铁缺乏贫血

[0156] 此实施例测试如下的假设，即在一些情况中可以在比目前基于扩充细胞群体可能的时间早至少 90 天预测代偿性或隐性 IDA，所述细胞比均值更接近原点沿着 MCHC 线投影。在图 5 中描绘投影操作。首先，测定具有体积 (v) 和血红蛋白 (h) 的每个细胞沿着 MCHC 线的投影位置 (u)：

[0157] $u = v \cdot \cos\theta - h \cdot \sin\theta$

[0158] $$\theta = -\tan^{-1}\left(\frac{\bar{h}}{\bar{v}}\right)$$

[0159] 沿着 MCHC 线的阈值定义为平均投影位置 $\bar{u}$ 的比例 $(\varphi)$：$\varphi \cdot \bar{u}$。投影细胞分数位于原点间，并且然后如下计算此阈值，其中 $f_{MCHC}$ 是为 MCHC 线上的位置函数的投影细胞的概率密度：

[0160]

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

[0161] 通过比较图 2 中使用的稳态 CBC 的不同阈值的区别效率选择 $\varphi = 85\%$ 阈值。在与

其它阈值（包括 70%、75%（图 8C 中显示）、80%、90%（图 8D）、和 100%）相比时，图 8A-B 中显示的 85% 阈值提供最大分开。基于此训练集选择 P0.85 阈值数值 0.121。

[0162] 然后，鉴定 40 个新的且独立的患者 CBC，它们都是正常的。这些正常 CBC 中的 20 个来自 30-90 天后具有别的正常 CBC 的个体，而这些正常 CBC 中的 20 个来自在不超过 90 天后表现出 IDA 的个体。排除在两个 CBC 间具有急性出血或任何铁补充的患者。上文在实施例 1 中列出了 IDA 的定义。图 3A-D 显示了 P0.85 阈值 0.121 能够以 75% 的灵敏度和 100% 的特异性在此独立测试组中预测 IDA。

[0163] 实施例 4：小红细胞性贫血的区别诊断

[0164] 为了评估 $D_h$ 在区别两种最常见的小红细胞性贫血原因中的诊断精确性，参数适合于 10 例训练病例：具有 IDA 的 5 例和具有 TT 的 5 例，如图 4 中所显示的。上文在实施例 2 中提供了病例定义。TT 训练组具有范围为 $1.7x10^{-15}$ 至 $2.3x10^{-15}$ 的 $D_h$。IDA 训练组具有范围为 0.009 至 0.043 的 $D_h$。阈值数值等于 0.0045，选择 IDA 训练组间的最低 $D_h$ 和 TT 训练组的最高值的平均值。然后，鉴定 50 例新的且独立的病例，患有 IDA 的 22 例和患有 TT 的 28 例。将所有 22 例 IDA 病例正确分类，并正确分类 27/28TT 病例以实现总体诊断精确性 98%。此精确性优于 4 种通常引用的判别式函数的 (20)：Green 和 King 公式产生 92% 的精确性，Micro/Hypo 产生 84% 的精确性，Mentzer 产生 68% 的精确性，并且 England 和 Fraser 产生 57% 的精确性。关于这些其它判别式函数的更多详情，参见参考文献 (20)。

[0165] 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>.

[0166] 2.Donofrio G,et al.(1995)Simultaneous Measurement of Reticulocyte and Red-Blood-Cell Indexes in Healthy-Subjects and Patients with Microcytic Anemia. Blood85(3):818-823.

[0167] 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 Haematology135(3):395-404.

[0168] 4.Waugh RE,et al.(1992)Rheologic Properties of Senescent Erythrocytes-Loss of Surface-Area and Volume with Red-Blood-Cell Age. Blood79(5):1351-1358.

[0169] 5.Willekens FL,et al.(2003)Hemoglobin loss from erythrocytes in vivo results from spleen-facilitated vesiculation.Blood101(2):747-751.

[0170] 6.Sens P&Gov N(2007)Force balance and membrane shedding at the red-blood-cell surface.Phys Rev Lett98(1):018102.

[0171] 7.Willekens FL,et al.(2005)Liver Kupffer cells rapidly remove red blood cell-derived vesicles from the circulation by scavenger receptors. Blood105(5):2141-2145.

[0172] 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.Blood86(1):334-341.

[0173] 9.Zenker S,Rubin J,&Clermont G(2007)From inverse problems in mathematical physiology to quantitative differential diagnoses.PLoSComput. Biol.3(11):2072-2086.

[0174] 10.Zwanzig R(2001)Nonequilibrium statistical mechanics(Oxford University Press,Oxford;New York)pp ix,222p.

[0175] 11.KampenNGv(1992)Stochastic processes in physics and chemistry(North-Holland,Amsterdam;New York)Rev.and enl.Ed pp xiv,465p.

[0176] 12.Lang KS,et al.(2005)Mechanisms of suicidal erythrocyte death. Cellular Physiology and Biochemistry15(5):195-202.

[0177] 13.Garner C,et al.(2000)Genetic influences on F cells and other hematologic variables:a twin heritability study.Blood95(1):342-346.

[0178] 14.Robbins SL,Kumar V,&Cotran RS(2010)Robbins and Cotran pathologic basis of disease(Saunders/Elsevier,Philadelphia,PA)8th Ed pp xiv,1450p.

[0179] 15.Milbrandt EB,et al.(2006)Predicting late anemia in critical illness.Crit.Care10(1).

[0180] 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.

[0181] 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.

[0182] 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.

[0183] 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 Health40(3):576-580.

[0184] 20.Ntaios G,et al.(2007)Discrimination indices as screening tests for beta-thalassemic trait.Ann.Hematol.86(7):487-491.

[0185] 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 A120hematologic analyzer.J.Lab.Clin.Med.146(6):333-340.

[0186] 其它实施方案

[0187] 应当理解，虽然本发明已经结合其详细描述进行了描述，但是前述描述意图例示而并不限制本发明的范围，其以所附权利要求书的范围限制。其它方面、优点、和修改在所附权利要求书的范围内。

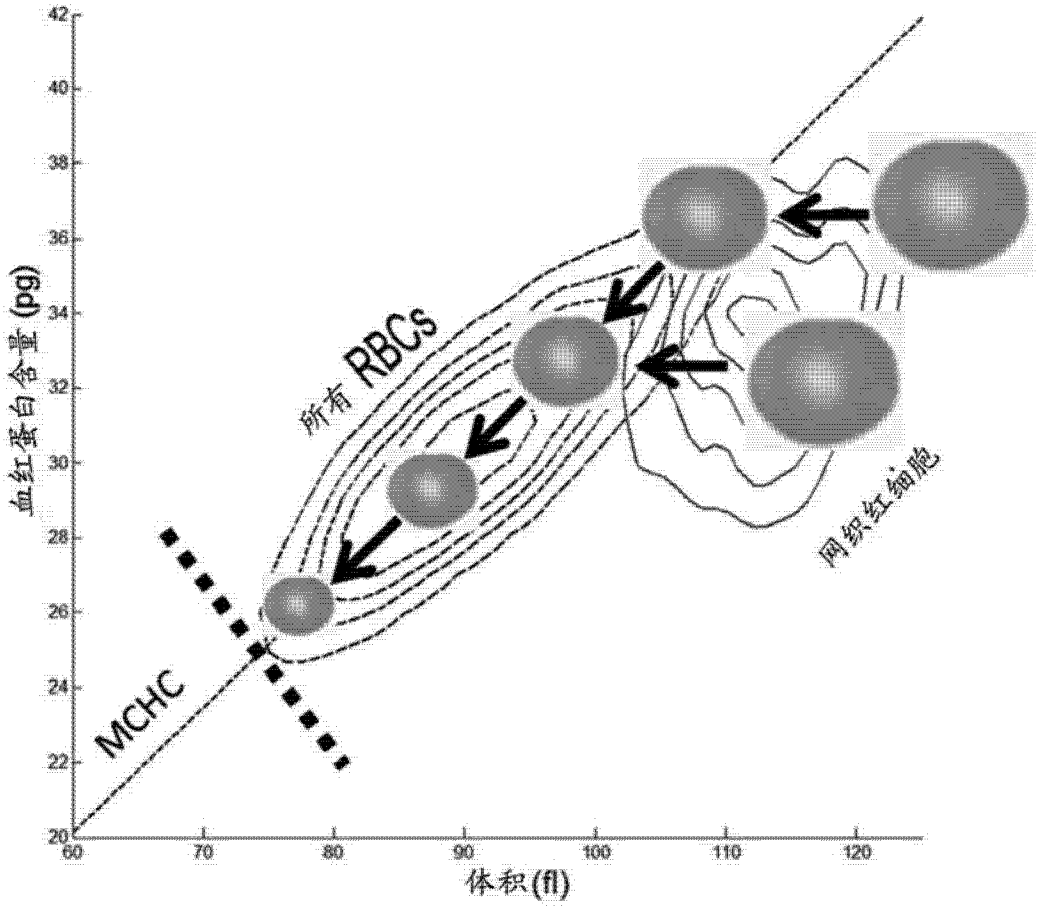

血红蛋白含量 (pg)
体积(fl)
所有 RBCs
网织红细胞
MCHC
42
40
38
36
34
32
30
28
26
24
22
20
60
70
80
90
100
110
120

图 1A

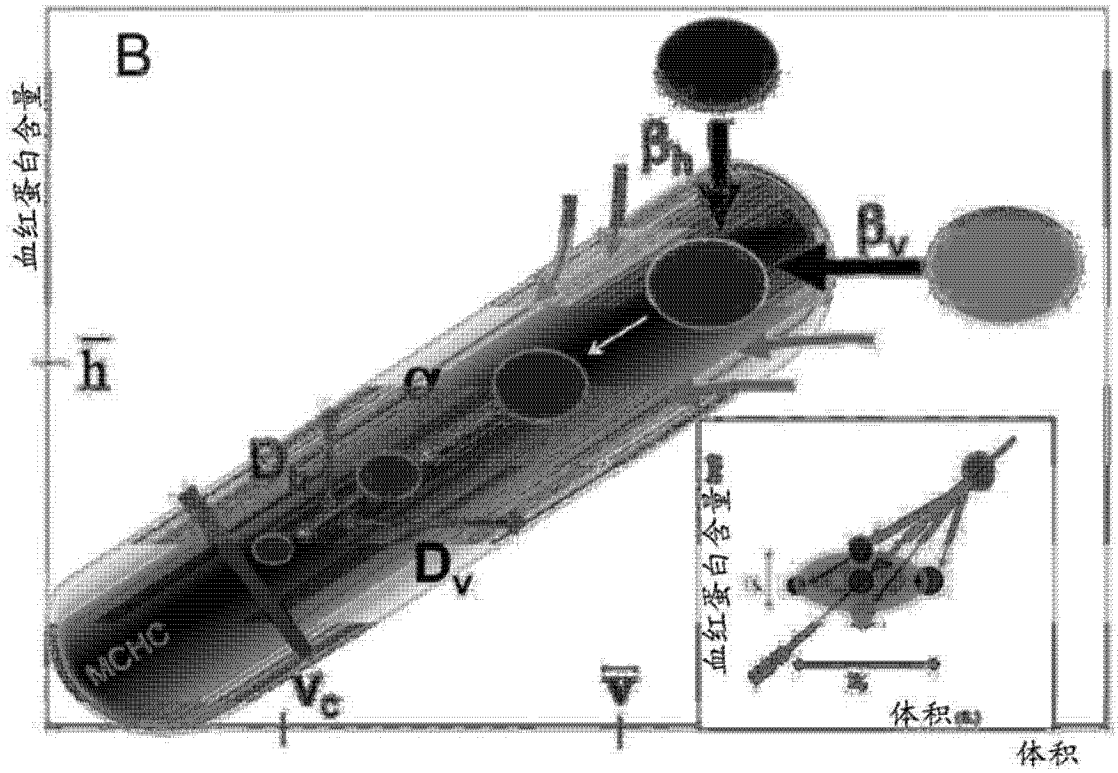



图 1B

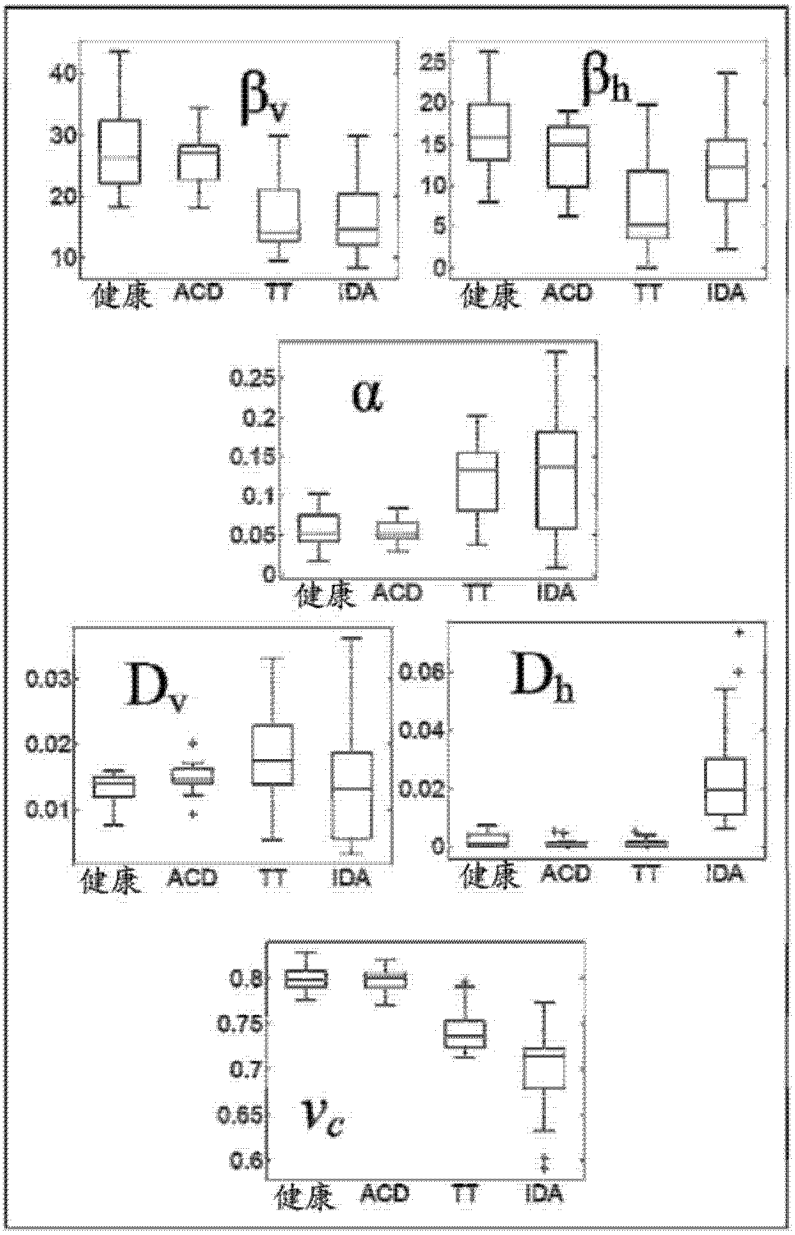

βv
βh
α
Dv
Dh
vc
健康
ACD
TT
IDA

图 2

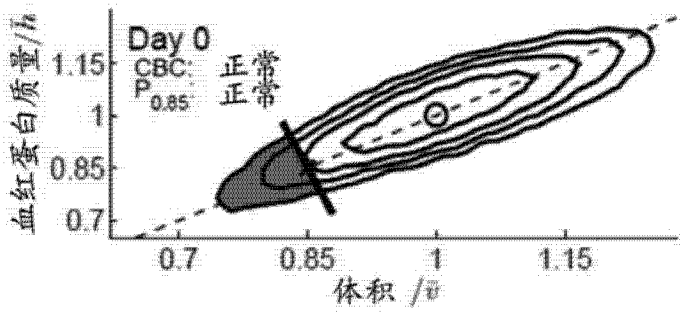

Day 0
CBC:
P0.85
正常
正常
血红蛋白质量/h
体积 /v
0.7
0.85
1
1.15

图 3A

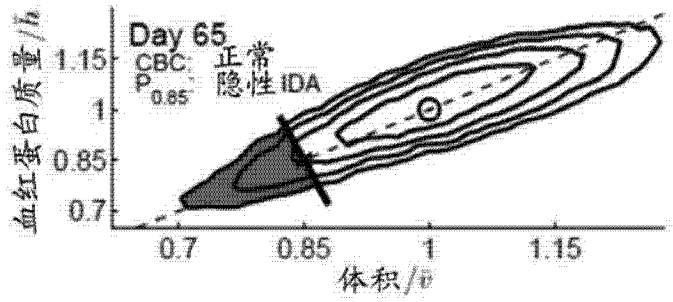

Day 65
CBC: 正常
$P_{0.85}$: 隐性IDA
血红蛋白质量/h
1.15
1
0.85
0.7
0.7
0.85
1
1.15
体积/v

图 3B

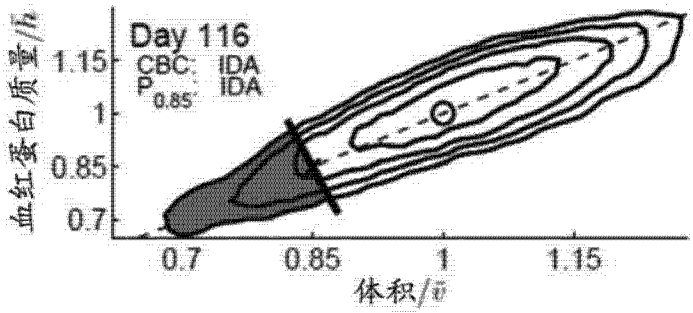

Day 116
CBC: IDA
$P_{0.85}$: IDA
血红蛋白质量/h
1.15
1
0.85
0.7
0.7
0.85
1
1.15
体积/v

图 3C

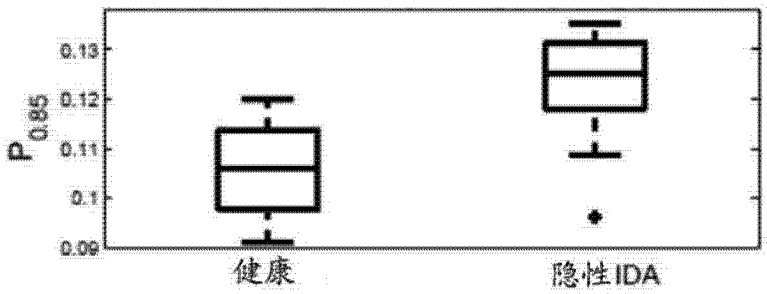

$P_{0.85}$
0.13
0.12
0.11
0.1
0.09
健康
隐性IDA

图 3D

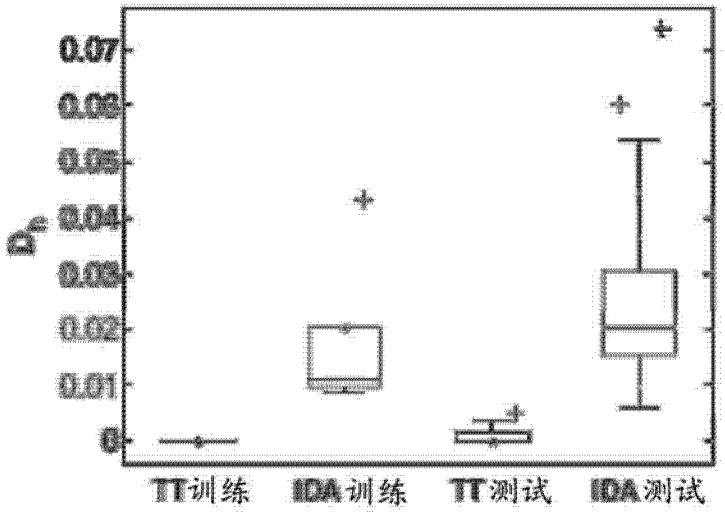

0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
TT训练
IDA训练
TT测试
IDA测试

图 4

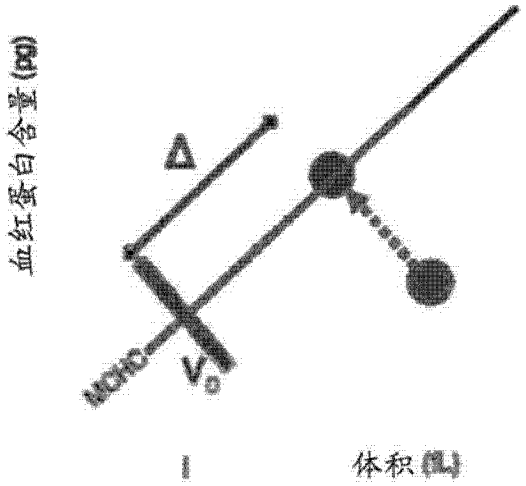

血红蛋白含量(g)
Δ
MCHC
$V_0$
体积(L)

图 5

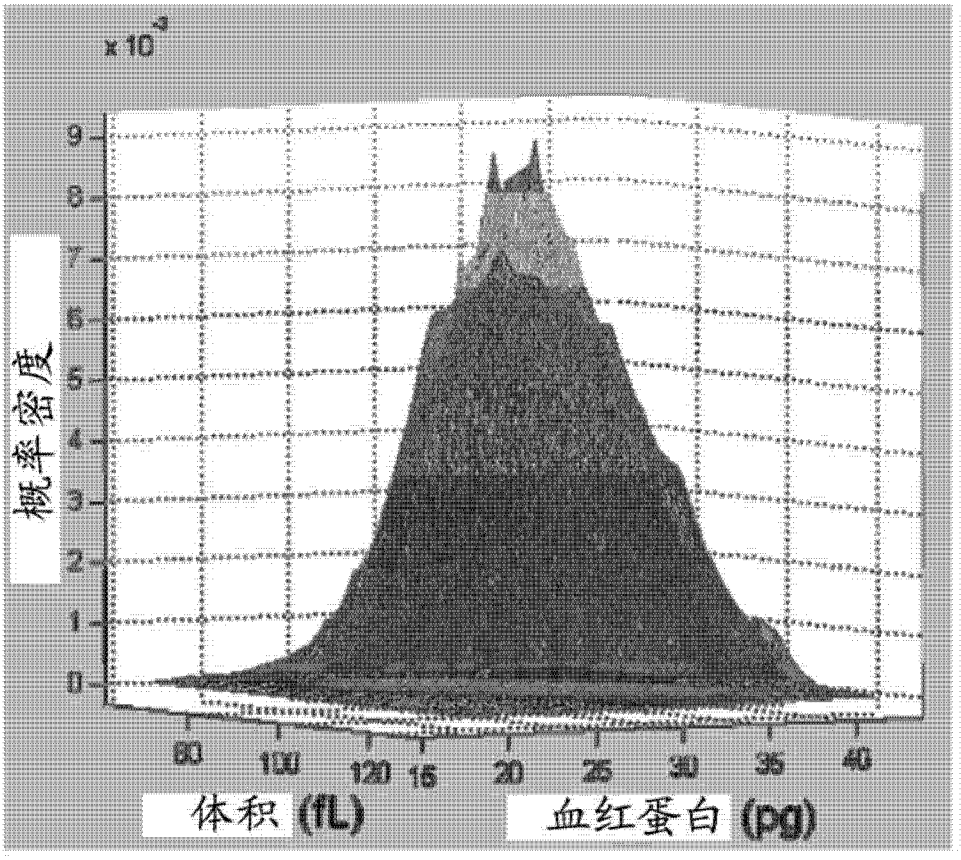



图 6A

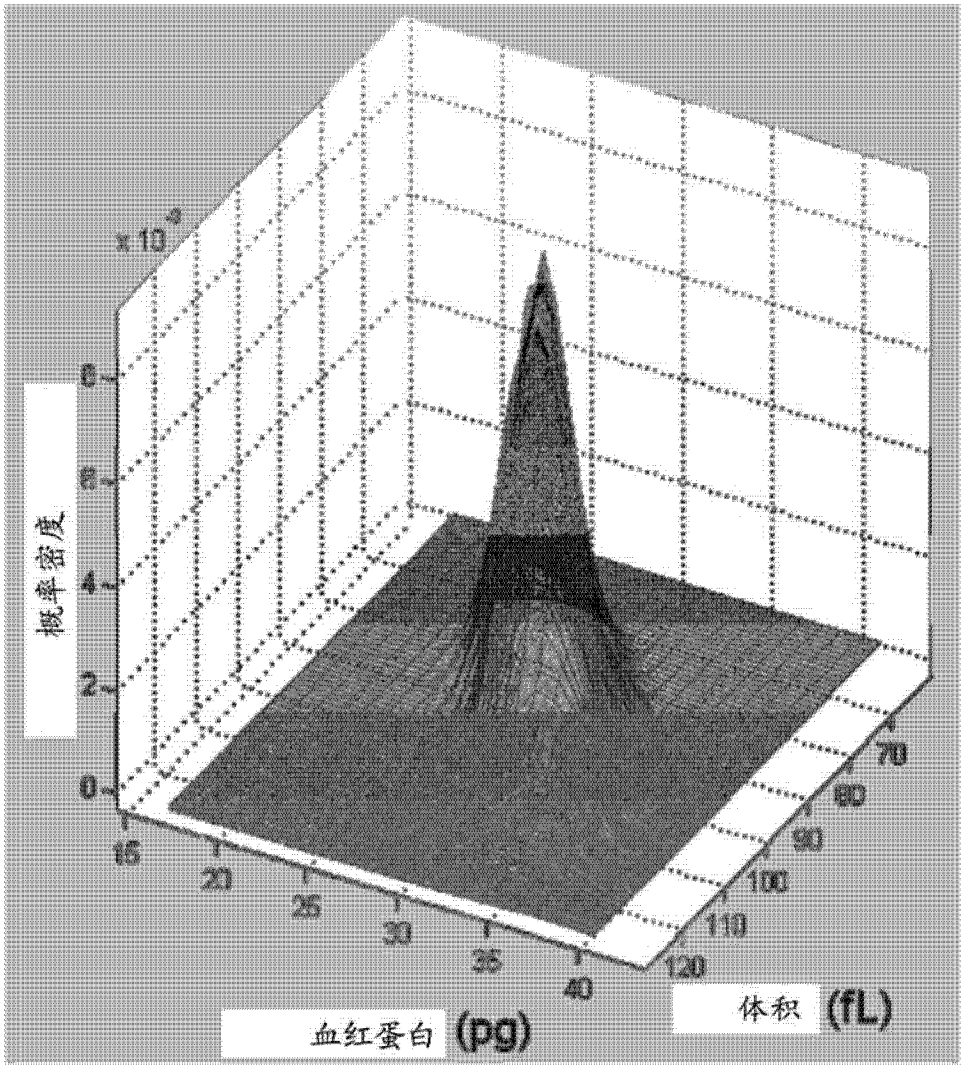

x 10 -3
概率密度
0
2
4
6
8
15
20
25
30
35
40
血红蛋白 (pg)
70
80
90
100
110
120
体积 (fL)

图 6B

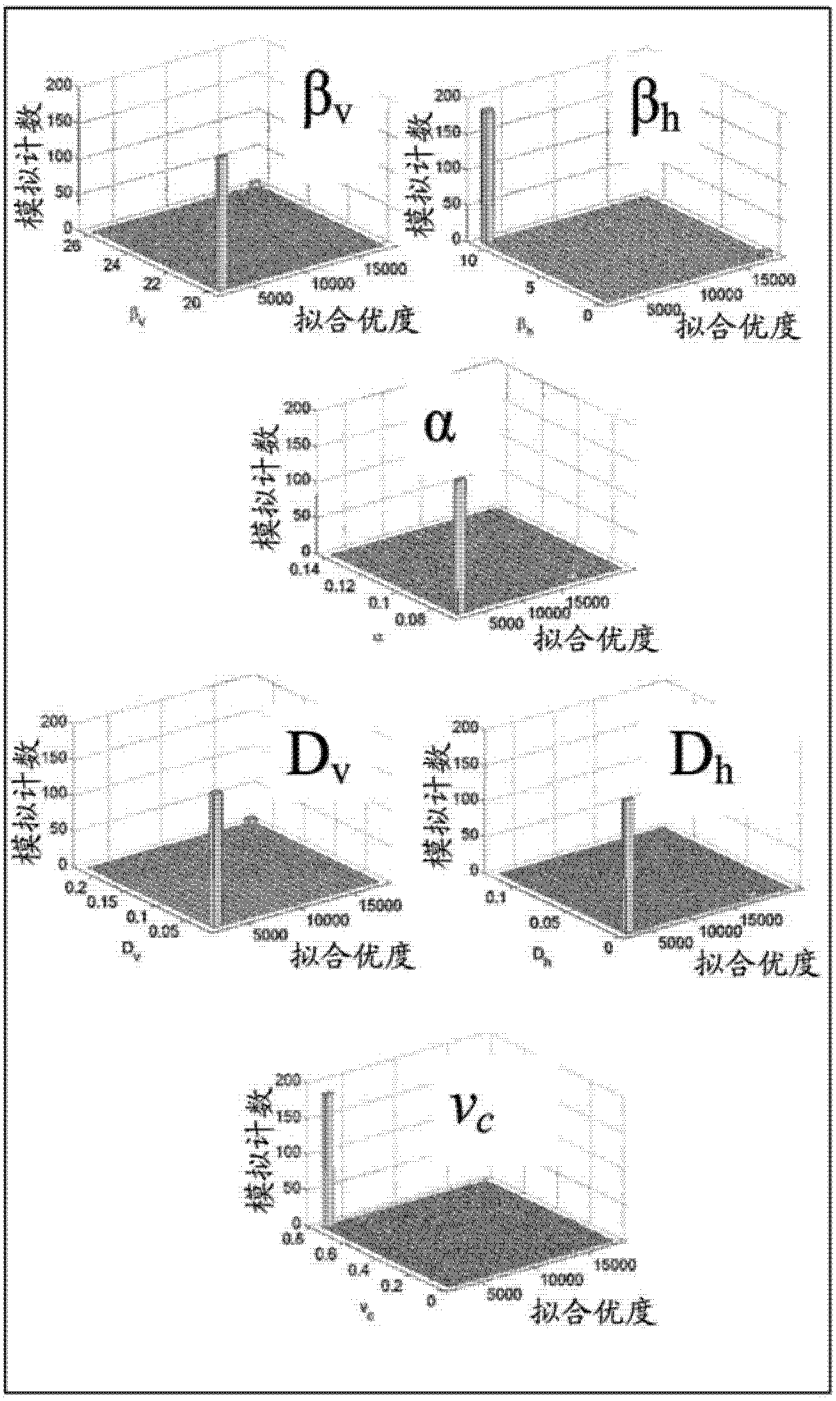

βv
模拟计数
拟合优度
βh
模拟计数
拟合优度
α
模拟计数
拟合优度
Dv
模拟计数
拟合优度
Dh
模拟计数
拟合优度
vc
模拟计数
拟合优度

图 7

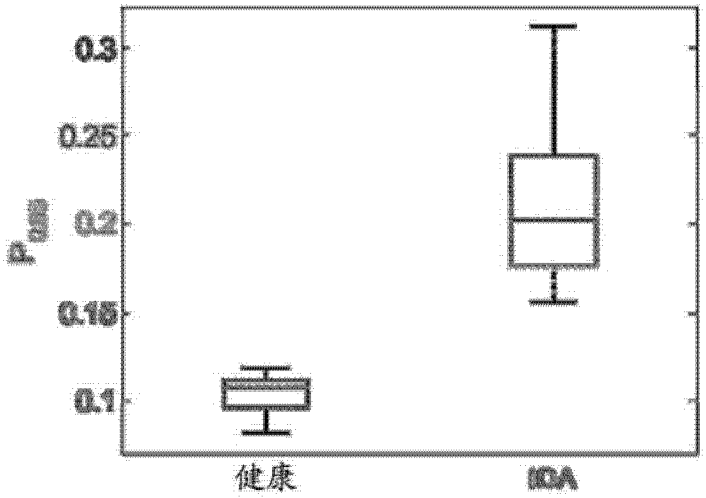

0.3
0.25
0.2
0.15
0.1
健康
IDA

图 8A

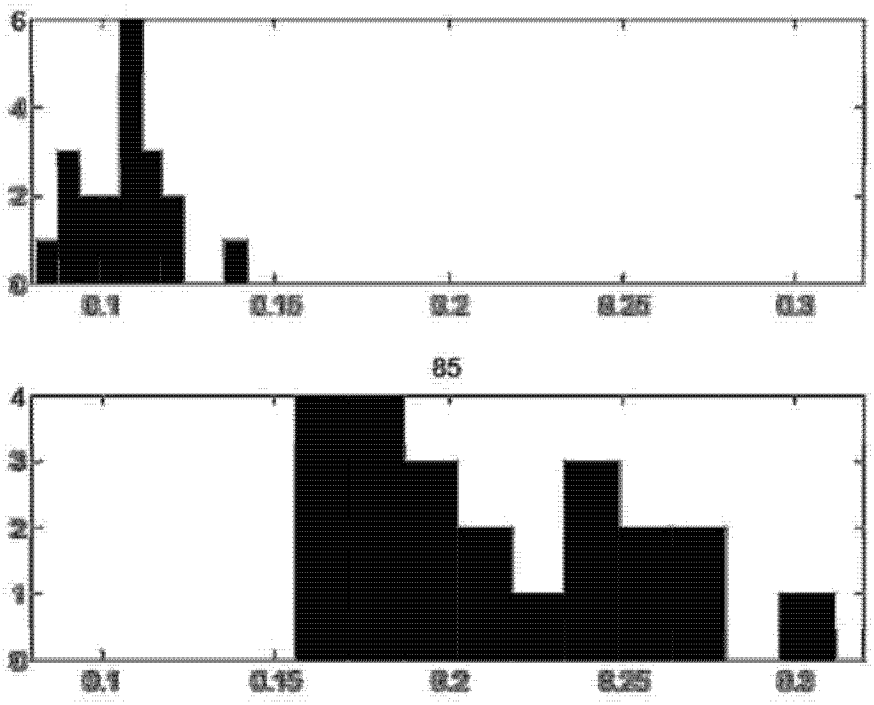

6
4
2
0
0.1
0.15
0.2
0.25
0.3
85
4
3
2
1
0
0.1
0.15
0.2
0.25
0.3

图 8B

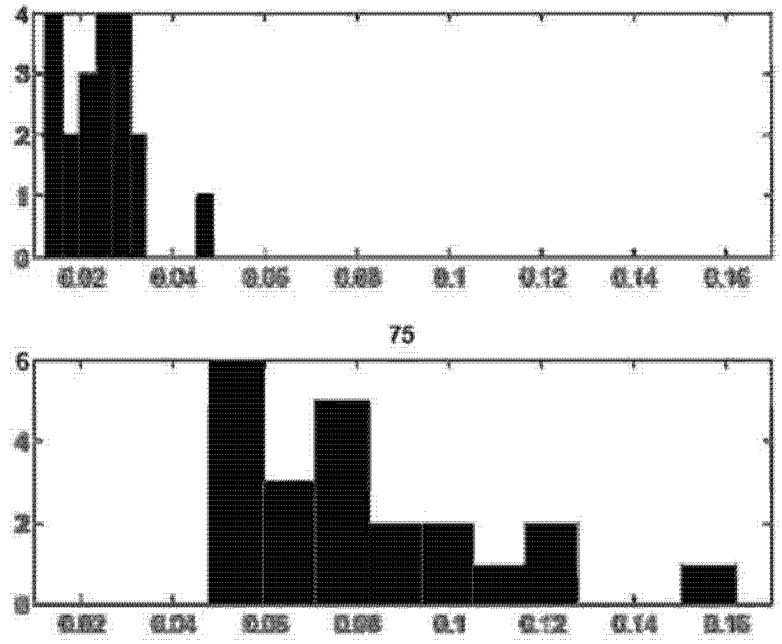



图 8C

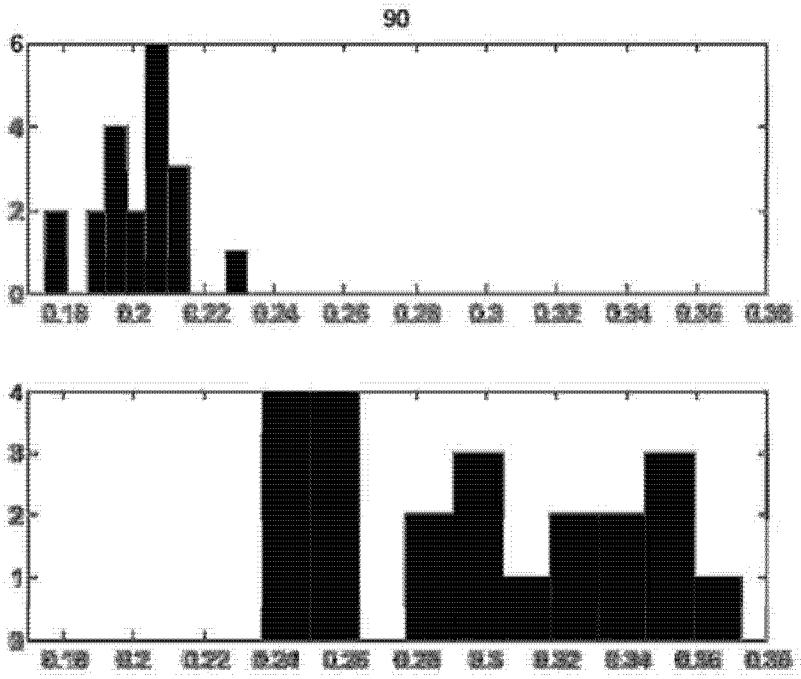



图 8D

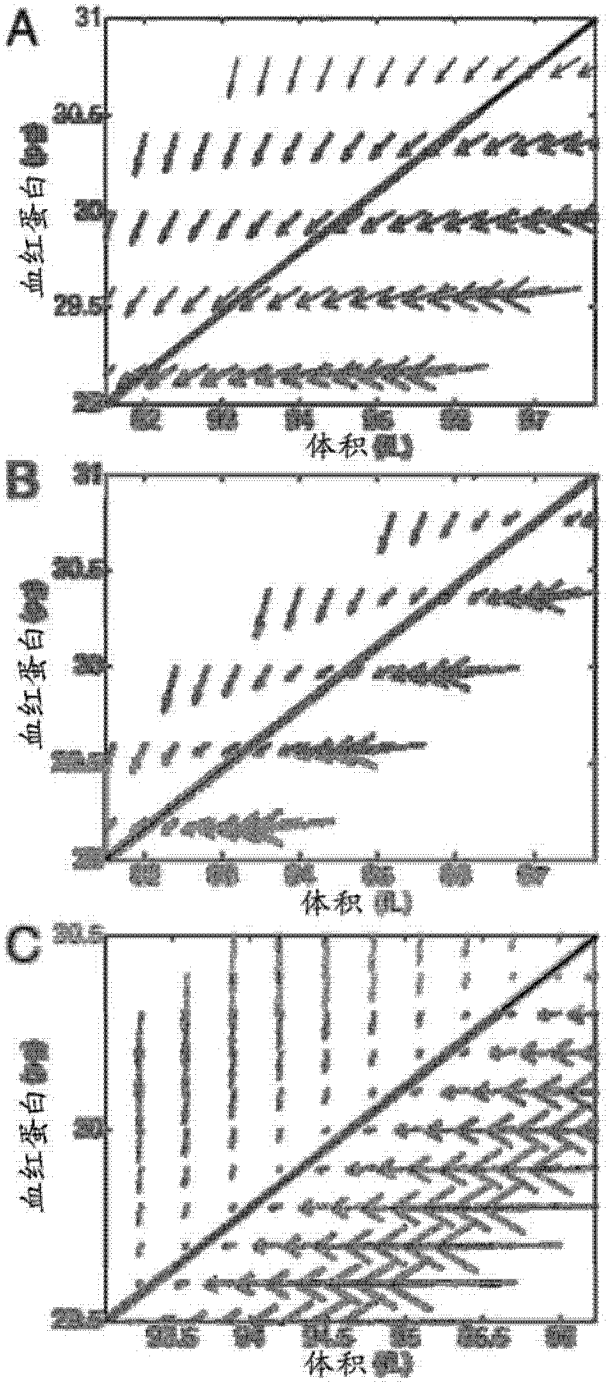



图 9A-C

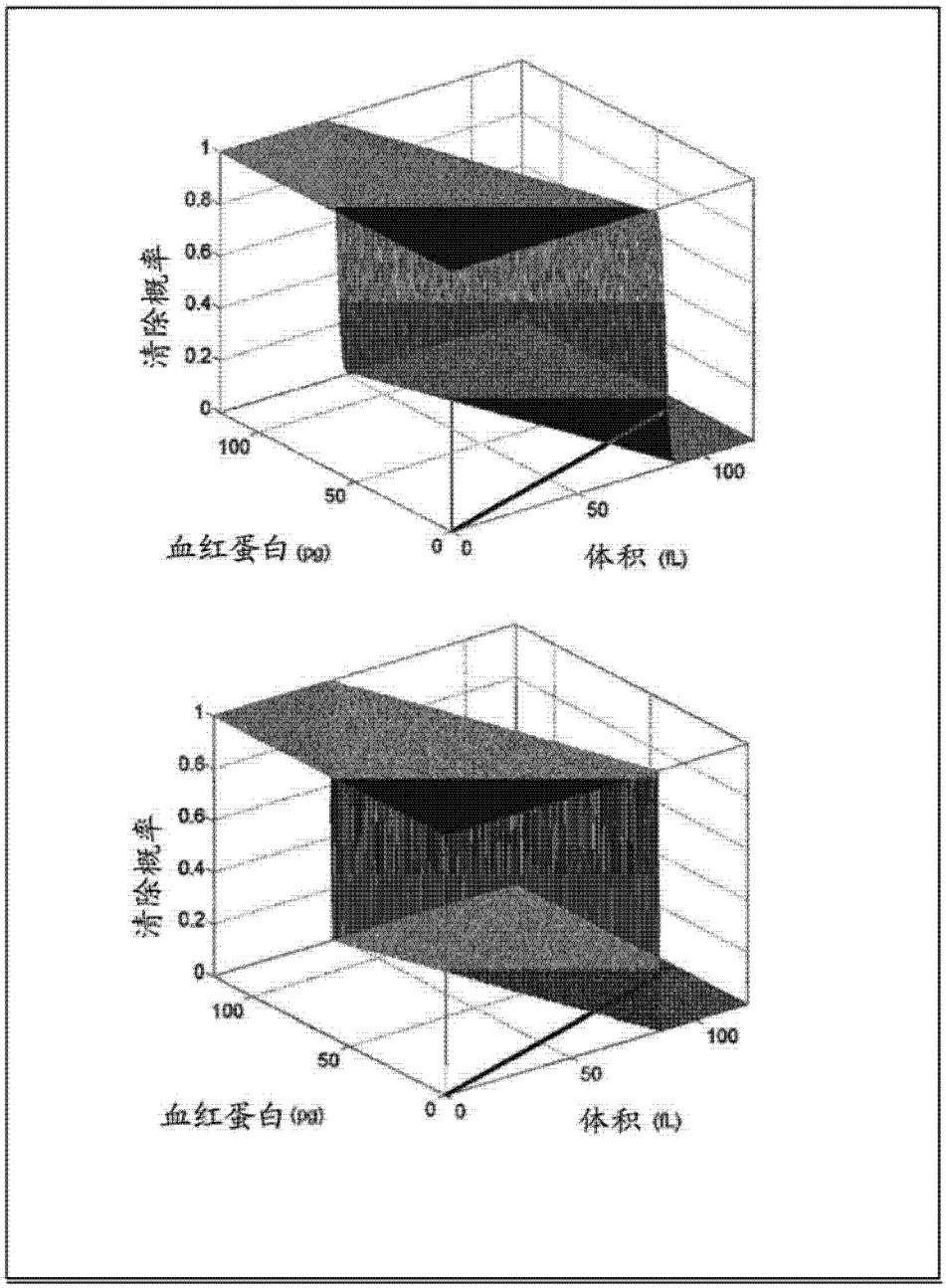

清除概率
1
0.8
0.6
0.4
0.2
0
100
50
0
血红蛋白 (pg)
0
50
100
体积 (fL)
清除概率
1
0.8
0.6
0.4
0.2
0
100
50
0
血红蛋白 (pg)
0
50
100
体积 (fL)

图 10A-B

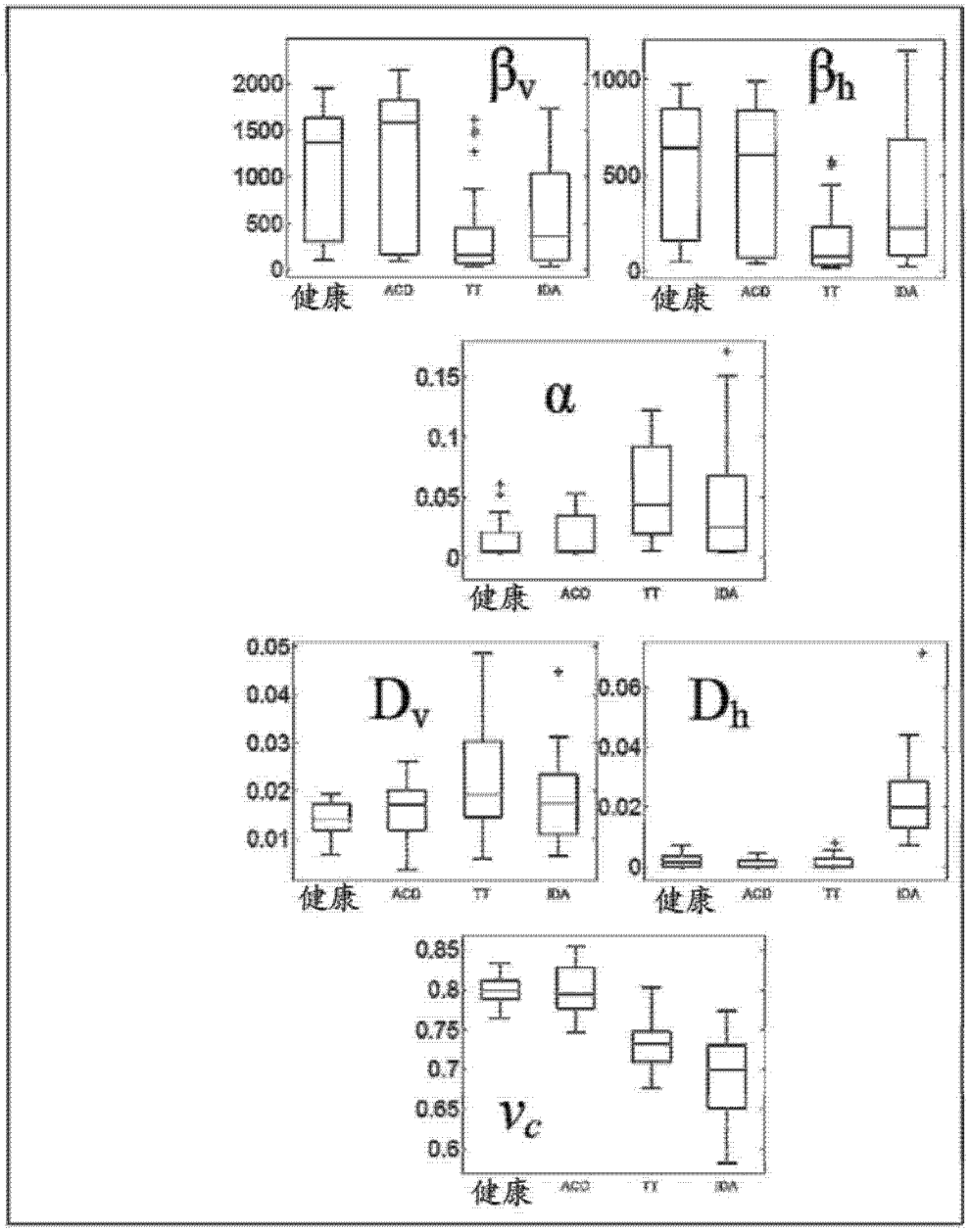

βv
2000
1500
1000
500
0
健康
ACD
TT
IDA
βh
1000
500
0
α
0.15
0.1
0.05
Dv
0.05
0.04
0.03
0.02
0.01
Dh
0.06
0.04
0.02
vc
0.85
0.8
0.75
0.7
0.65
0.6

图 11

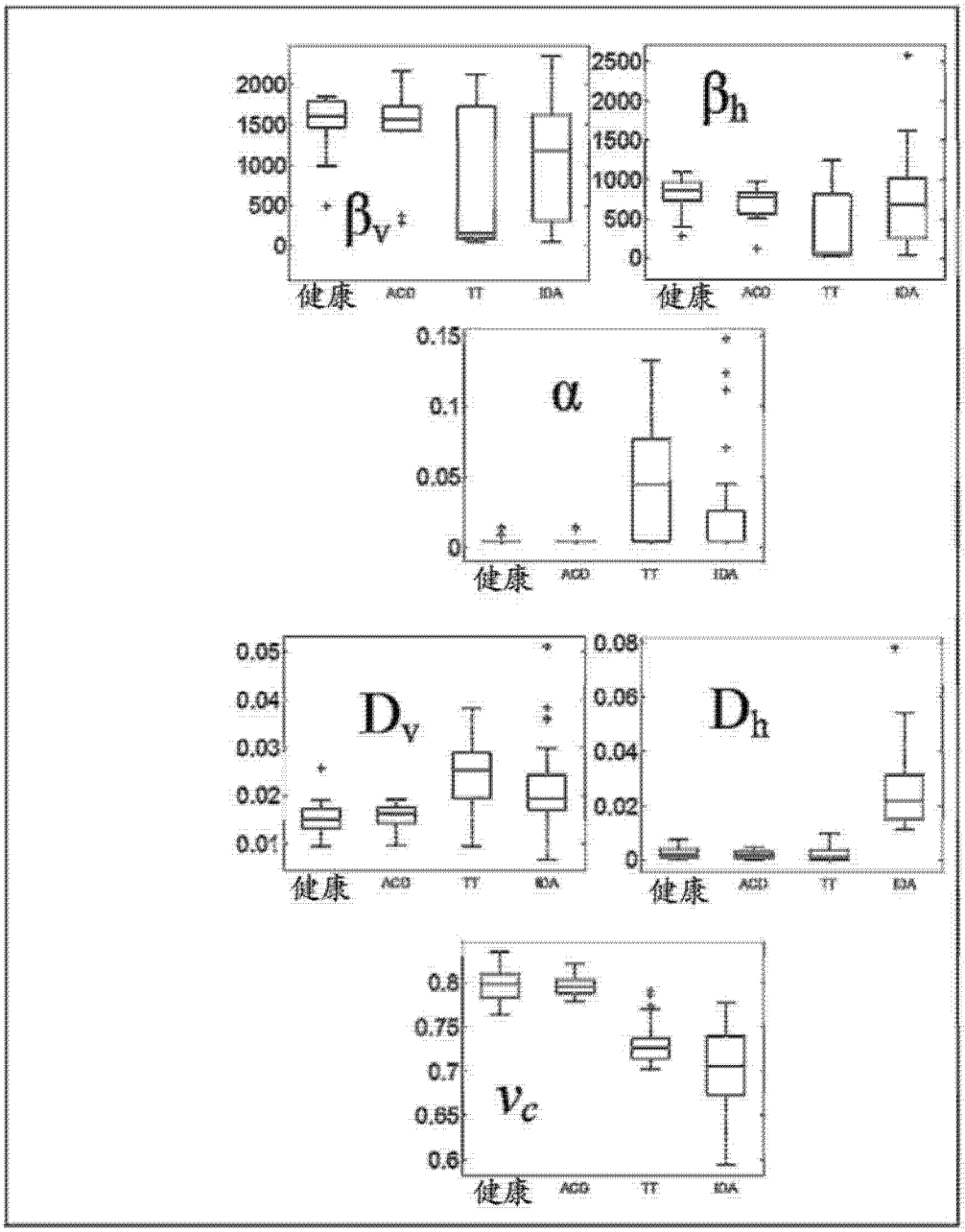

2000
1500
1000
500
0
$\beta_v$
健康
ACD
TT
IDA
2500
2000
1500
1000
500
0
$\beta_h$
健康
ACD
TT
IDA
0.15
0.1
0.05
0
α
健康
ACD
TT
IDA
0.05
0.04
0.03
0.02
0.01
$D_v$
健康
ACD
TT
IDA
0.08
0.06
0.04
0.02
0
$D_h$
健康
ACD
TT
IDA
0.8
0.75
0.7
0.65
0.6
$v_c$
健康
ACD
TT
IDA

图 12

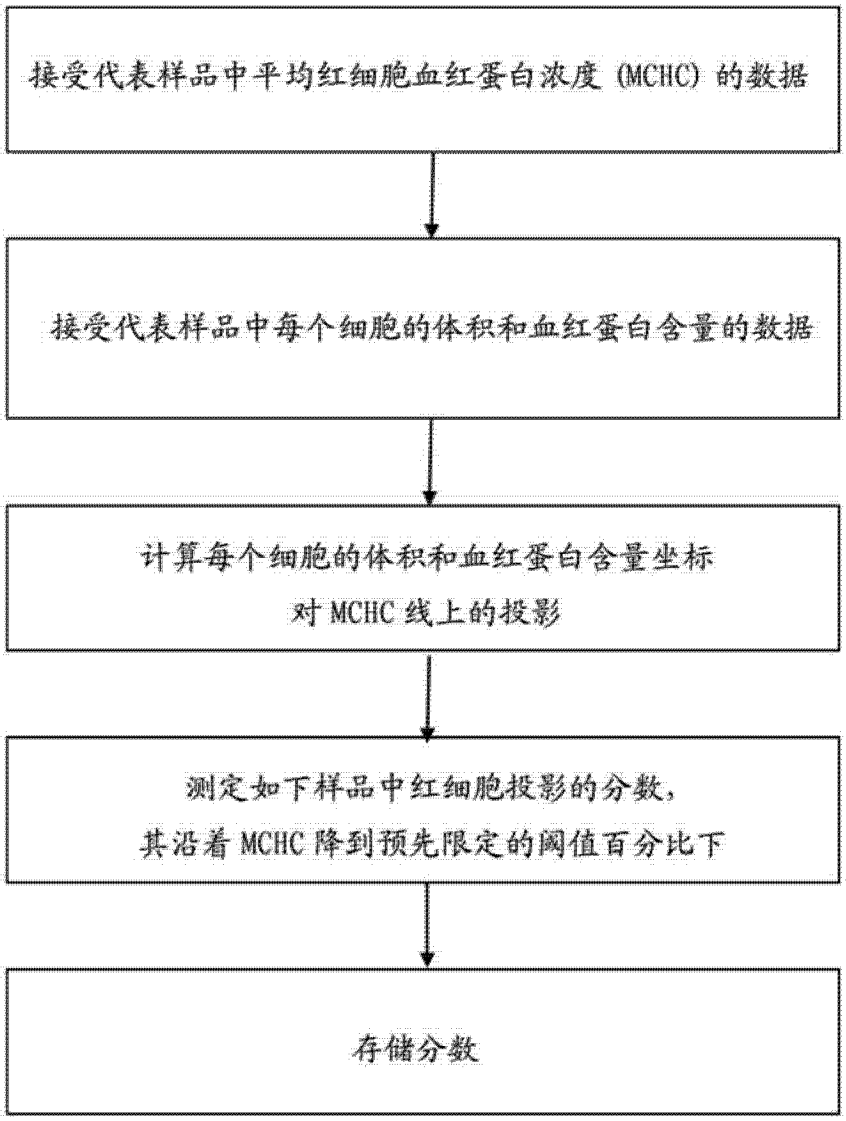

接受代表样品中平均红细胞血红蛋白浓度 (MCHC) 的数据
接受代表样品中每个细胞的体积和血红蛋白含量的数据
计算每个细胞的体积和血红蛋白含量坐标
对 MCHC 线上的投影
测定如下样品中红细胞投影的分数，
其沿着 MCHC 降到预先限定的阈值百分比下
存储分数

图 13A

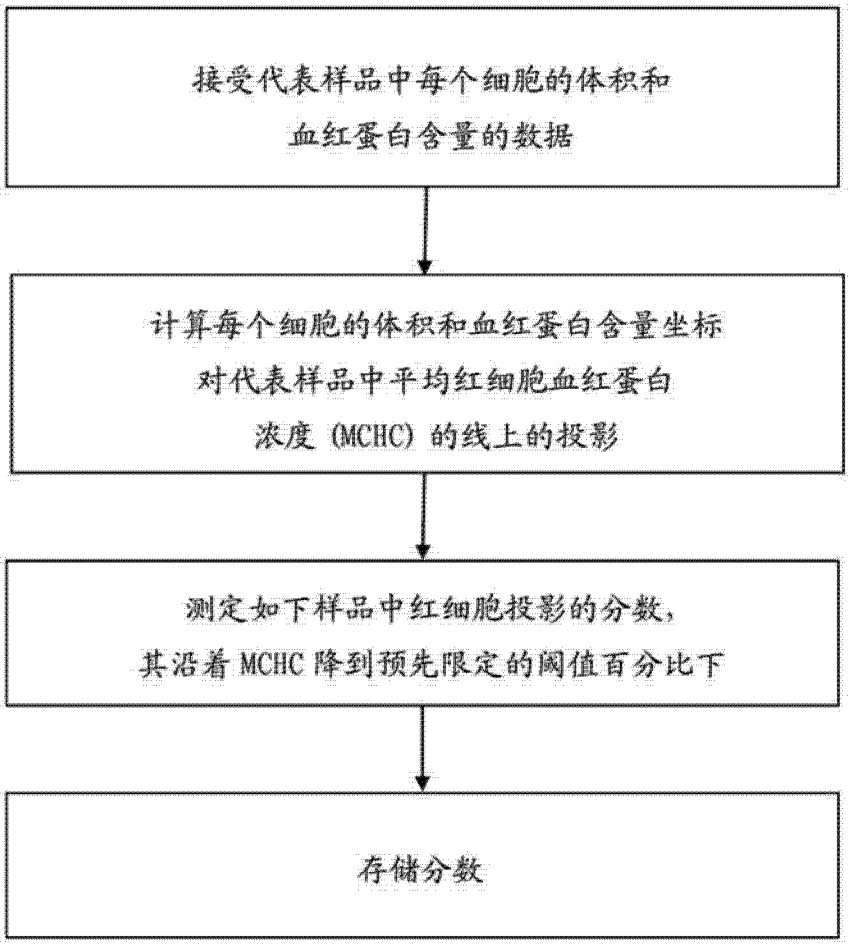

接受代表样品中每个细胞的体积和
血红蛋白含量的数据
计算每个细胞的体积和血红蛋白含量坐标
对代表样品中平均红细胞血红蛋白
浓度 (MCHC) 的线上的投影
测定如下样品中红细胞投影的分数，
其沿着 MCHC 降到预先限定的阈值百分比下
存储分数

图 13B

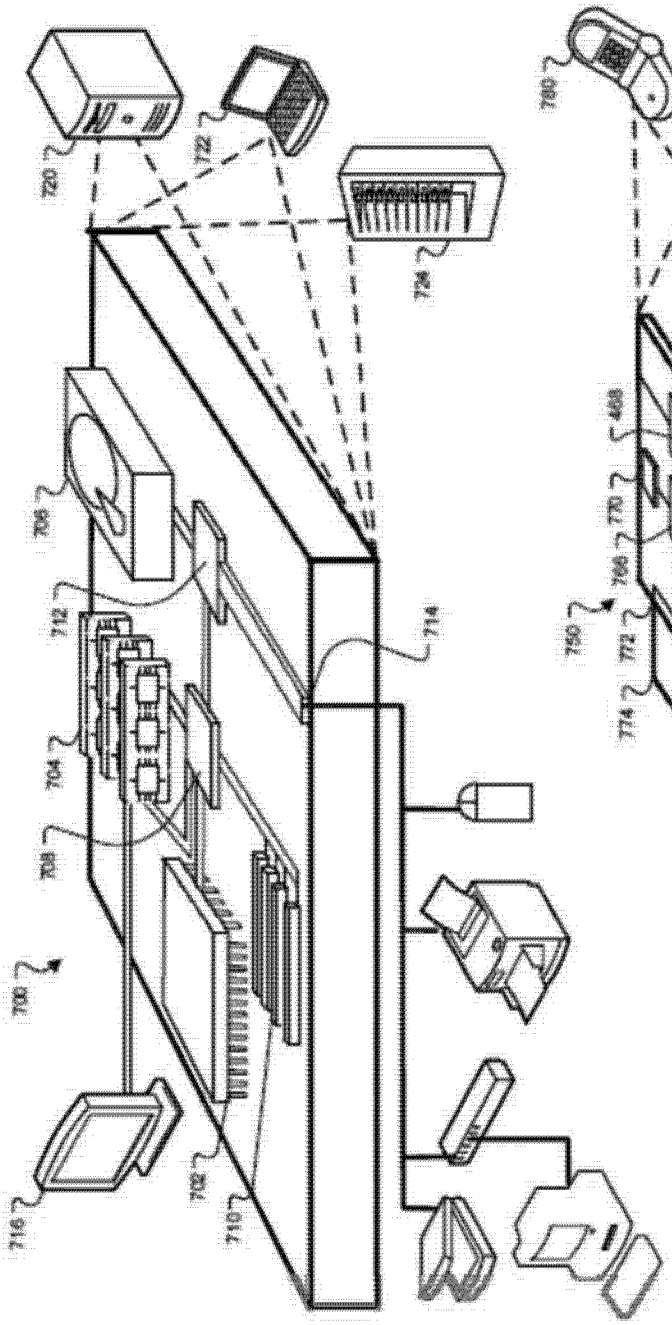

720
722
726
706
712
714
704
708
700
716
702
710

图 14

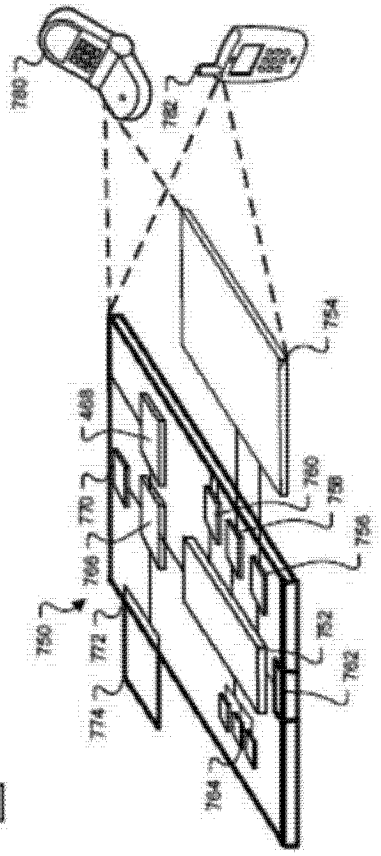

760
762
754
468
770
766
750
772
774
764
752
756
758

图 7

```
function r = modelSample(sAccession, bDB, bParallel, vSimID, nStartingPoints,
vMuAge)
    % Given an accession, calculate optimal model parameters and store
    % them.
    r = 0;
    if (nargin == 1)
        bDB = 1;
        bParallel = 0;
        vSimID = [72];
        nStartingPoints = 10;
        vMuAge = [50 25 40 60 75];
    end
    rand('twister', sum(100*clock));

    % If running in parallel with more than 1 starting point, try to
    % execute all starting points in parallel at the same time.
    if (bParallel)
        if (nStartingPoints > 1)
            jm = findResource('scheduler','type','lsf');
            set(jm,'ClusterMatlabRoot','/opt/matlab');
            set(jm, 'SubmitArguments', '-q all_12h');
            job=createJob(jm);
        else

            if (~matlabpool('size'))
                matlabpool local
            end
        end
    end

    % And for each set of simulation parameters.
    stSimulationParameters = getSimulationParameters(bDB, '7/01/09', vSimID);

    % Get the steady state values for this accession.
    stSteadyState = getSteadyState(bDB, sAccession, '7/01/09',
stSimulationParameters);
%return
    % For each starting point.
    stStartingPoints = getStartingPoints(bDB, '7/01/09', stSimulationParameters,
nStartingPoints);

    for ixSimParams = 1:length(stSimulationParameters)
        for ixMuAge = 1:length(vMuAge)
            stSteadyState.muAge = vMuAge(ixMuAge);
            vStart = randperm([length(stStartingPoints)]);
            for (ixStart = 1:length(vStart))
                if (bParallel && (nStartingPoints > 1))
                    csInputs = {stSteadyState; stStartingPoints(vStart(ixStart));
stSimulationParameters(ixSimParams); bDB; sAccession; bParallel};
                    createTask(job, @findOptimalParams, 1, csInputs);
                else
                    % Find the optimal parameters.
                    stParameters = findOptimalParams(stSteadyState,
stStartingPoints(vStart(ixStart)), stSimulationParameters(ixSimParams), bDB,
sAccession, bParallel);
                end
            end
        end
    end
    if (bParallel && (nStartingPoints > 1))
        submit(job);
```

图 15

```
    end
end

function stSteadyState = getSteadyState(bDB, sAccession, sDate,
stSimulationParameters)
    stSteadyState = [];
    if (bDB)
        stSteadyState = queryFCSSteadyState(sAccession, sDate,
stSimulationParameters(1).nSteadyStateVersion);
        %return
    else
        stSteadyState = readFCSSteadyState(sAccession, sDate,
stSimulationParameters(1).nSteadyStateVersion);
    end

    assert(~any([diff([stSimulationParameters(:).nMinV])
diff([stSimulationParameters(:).nMeshWidth])
diff([stSimulationParameters(:).nMaxV])]));
    assert(~any([diff([stSimulationParameters(:).nMinH])
diff([stSimulationParameters(:).nMeshWidth])
diff([stSimulationParameters(:).nMaxH])]));

    [V,H] =
meshgrid([stSimulationParameters(1).nMinV:stSimulationParameters(1).nMeshWidth:st
SimulationParameters(1).nMaxV], ...

[stSimulationParameters(1).nMinH:stSimulationParameters(1).nMeshWidth:stSimulatio
nParameters(1).nMaxH]);
    H = flipud(H);
    vEmpirical = strcmp({stSimulationParameters(:).sInitialDistribution}',
'Empirical');
    bEmpirical = all(vEmpirical);
    if (bEmpirical)
        [Phat, P, vi, hi] = makeEmpiricalPDF(stSteadyState.rVData,
stSteadyState.rHCData);
        P_0 = interp2(vi, hi, Phat, V, H, 'linear', 0);
    else
        sdV = (stSteadyState.rMCV*stSteadyState.rRDW);
        sdH = (stSteadyState.rHDW*stSteadyState.rMCH);
        nCovar = stSteadyState.rRho*sdV*sdH;
        mSigma = [sdV^2 nCovar; nCovar sdH^2];
        P_0 = mvnpdf([V(:) H(:)], [stSteadyState.rMCV stSteadyState.rMCH],
mSigma);

        sdV = (stSteadyState.stAdviaFile.MCV*stSteadyState.stAdviaFile.rRDW);
        sdH = (stSteadyState.stAdviaFile.HDW*stSteadyState.stAdviaFile.MCH);
        nCovar = stSteadyState.rho*sdV*sdH;
        mSigma = [sdV^2 nCovar; nCovar sdH^2];
        [vData] = mvnrnd([stSteadyState.stAdviaFile.MCV
stSteadyState.stAdviaFile.MCH], mSigma, numel(stSteadyState.vData));
        stSteadyState.vData = vData(:,1);
        stSteadyState.hcData = 100*vData(:,2)./vData(:,1);
    end
    P_0 = P_0(:)/sum(P_0(:));
    stSteadyState.P_0 = P_0;
end

function stSteadyState = readFCSSteadyState(sAccession, sDate, nVersion)
    if (nargin < 3)
        nVersion = 0;
    end
```

图 15 续

```
% Read the sample XLS file to get the Advia File ID.
[num, txt, raw] = xlsread('sample.xls');
ixRow = find(num(:,2) == str2num(sAccession));
idAdvia = num(ixRow, 1);

% Read the advia file XLS file to get the FCS location and the steady
% state measurements.
%[num, txt, raw] = xlsread('advia.xls');
%save('advia.mat','num','txt','raw');
load('advia.mat');
nOffset = size(txt, 1) - size(num, 1);
ixRow = find(num(:,2) == idAdvia, 1, 'last');
stAdviaFile.ID = num(ixRow, 1);
stAdviaFile.RawDataFile = txt{ixRow + nOffset, 3};
stAdviaFile.MasterFCSFile = txt{ixRow + nOffset, 4};
stAdviaFile.RBCFCSFile = txt{ixRow + nOffset, 5};
stAdviaFile.ReticFCSFile = txt{ixRow + nOffset, 6};
stAdviaFile.HCT = num(ixRow, 7);
stAdviaFile.MCV = num(ixRow, 8);
stAdviaFile.RDW = num(ixRow, 9);
stAdviaFile.MCH = num(ixRow, 10);
stAdviaFile.HDW = num(ixRow, 11);
stAdviaFile.rMCV = num(ixRow, 12);
stAdviaFile.rRDW = num(ixRow, 13);
stAdviaFile.rMCH = num(ixRow, 14);
stAdviaFile.rHDW = num(ixRow, 15);
stAdviaFile.rFraction = num(ixRow, 16);

% Read the calculated steady state information.
% [num, txt, raw] = xlsread('steadystate.xls');
% save('steadystate.mat','num','txt','raw');
load('steadystate.mat');
ixRow = find(num(:,3) == stAdviaFile.ID);

if (isfinite(nVersion))
    ixID = find(num(ixRow, 15) == nVersion, 1, 'last');
    num = num(ixRow(ixID), :);
    nOffset = size(txt, 1) - size(num, 1);
    ixRow = 1;
else
    ixRow = ixRow(end);
end

tSteadyState.ID = num(ixRow, 1);
tSteadyState.nDate = num(ixRow, 2);
tSteadyState.MCV = num(ixRow, 4);
tSteadyState.RDW = num(ixRow, 5);
tSteadyState.MCH = num(ixRow, 6);
tSteadyState.HDW = num(ixRow, 7);
tSteadyState.rho = num(ixRow, 8);
tSteadyState.rFraction = num(ixRow, 9);
tSteadyState.rMCV = num(ixRow, 10);
tSteadyState.rRDW = num(ixRow, 11);
tSteadyState.rMCH = num(ixRow, 12);
tSteadyState.rHDW = num(ixRow, 13);
tSteadyState.rRho = num(ixRow, 14);
tSteadyState.nSteadyStateVersion = num(ixRow, 15);
if (isnan(tSteadyState.nSteadyStateVersion))
    tSteadyState.nSteadyStateVersion = 0;
end

stAdviaFile = expandFCSFilenames(stAdviaFile);
```

图 15 续

```
    stAdviaFile.RBCFCSFile = mapFileName(stAdviaFile.RBCFCSFile);
    stAdviaFile.ReticFCSFile = mapFileName(stAdviaFile.ReticFCSFile);
    stAdviaFile.RawDataFile = mapFileName(stAdviaFile.RawDataFile);
    [nMCV, nRDW, nCHCM, nHDW, nRho, tSteadyState.vData, tSteadyState.hcData] =
getRBCPDF(stAdviaFile, tSteadyState.nSteadyStateVersion);
    [f, nMCV, nRDW, nMCH, nHDW, nRho, tSteadyState.rVData, tSteadyState.rHCData]
= getReticPDF(stAdviaFile,
tSteadyState.nSteadyStateVersion);
    tSteadyState.stAdviaFile = stAdviaFile;

    stSteadyState = tSteadyState;
end

function sNewName = mapFileName(sName)
    if (~exist('Z:\Research\Microcytosis'))
        if (~ispc())
            if (~isempty(sName))
                if (sName(1) ~= '\')
                    sName = ['~/research/Microcytosis/' sName];
                end
            end
            sNewName = strrep(sName, 'Z:\Research\Microcytosis',
'~/research/Microcytosis');
            sNewName = strrep(sName, 'C:\local\jhiggins\research\microcytosis',
'~/research/Microcytosis');
            sNewName = strrep(sNewName, '\', '/');
        else
            sNewName = strrep(sName, 'Z:\Research\Microcytosis', '.');
            if ((sName(1) ~= '\'))

            end
            sNewName = strrep(sNewName, '/', '\');
        end
    else
        sNewName = sName;
    end
end

function stStartingPoints = getStartingPoints(bDB, sDate, stSimulationParameters,
nStartingPoints)
    bLatinHypercube = 1;
    if (bLatinHypercube)
        stStartingPoints = 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.
    [num, txt, raw] = xlsread('startingpoint.xls');

    stStartingPoints = [];
    for ixStart = 1:size(num, 2)
        nDate = num(ixStart, 2);
        tStartingPoint.ID = num(ixStart, 1);
        tStartingPoint.nDate = nDate;
```

图 15 续

```
            tStartingPoint.alpha = num(ixStart, 3);
            tStartingPoint.betav = num(ixStart, 4);
            tStartingPoint.betah = num(ixStart, 5);
            tStartingPoint.Dv = num(ixStart, 6);
            tStartingPoint.Dh = num(ixStart, 7);
            tStartingPoint.v50 = num(ixStart, 8);
            stStartingPoints = [stStartingPoints; tStartingPoint];
        end
    end

    function stStartingPointsLatin = sampleLatinHypercube(stSimulationParameters,
    nStartingPoints)
        nDims = 6;
        nPoints = nStartingPoints;
        stStartingPointsLatin = [];

        vStartingPointsNarrow =
    strcmp({stSimulationParameters(:).sStartPointMethod}', 'Narrow');
        bNarrow = all(vStartingPointsNarrow);

        vExponential = strcmp({stSimulationParameters(:).sFunctionalForm}',
    'Exponential');
        bExponential = all(vExponential);
        vHybrid = strcmp({stSimulationParameters(:).sFunctionalForm}', 'Hybrid');
        bHybrid = all(vHybrid);
        vSimple = strcmp({stSimulationParameters(:).sFunctionalForm}', 'Simple');
        bSimple = all(vSimple);
        vAdditiveExponential = strcmp({stSimulationParameters(:).sFunctionalForm}',
    'Additive Exponential');
        bAdditiveExponential = all(vAdditiveExponential);
        vDecayingAdditiveExponential =
    strcmp({stSimulationParameters(:).sFunctionalForm}', 'Decaying Additive
    Exponential');
        bDecayingAdditiveExponential = all(vDecayingAdditiveExponential);

        if (bHybrid)
            if (~bNarrow)
                vRangeAlpha = [0 0.1];
                vRangeBetaV = [0 50];
                vRangeBetaH = [0 50];
                vRangeDV = [0 50];
                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

        mIX = lhsdesign(nPoints, nDims);
```

图 15 续

```
    for ix = 1:nPoints
        tStartingPoint.ID = -1;
        tStartingPoint.nDate = datenum(now);
        tStartingPoint.alpha = vRangeAlpha(1) + range(vRangeAlpha)*mIX(ix, 1);
        tStartingPoint.betav = vRangeBetaV(1) + range(vRangeBetaV)*mIX(ix, 2);
        tStartingPoint.betah = vRangeBetaH(1) + range(vRangeBetaH)*mIX(ix, 3);
        tStartingPoint.Dv = vRangeDV(1) + range(vRangeDV)*mIX(ix, 4);
        tStartingPoint.Dh = vRangeDH(1) + range(vRangeDH)*mIX(ix, 5);
        tStartingPoint.v50 = vRangeV50(1) + range(vRangeV50)*mIX(ix, 6);
        stStartingPointsLatin = [stStartingPointsLatin; tStartingPoint];
    end
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)
        stStartingPoints(ix).alpha =
bumpStartingPoint(stStartingPoints(ix).alpha, vRangeAlpha);
        stStartingPoints(ix).betav =
bumpStartingPoint(stStartingPoints(ix).betav, vRangeBetaV);
        stStartingPoints(ix).betah =
bumpStartingPoint(stStartingPoints(ix).betah, vRangeBetaH);
        stStartingPoints(ix).Dv = bumpStartingPoint(stStartingPoints(ix).Dv,
vRangeDV);
        stStartingPoints(ix).Dh = bumpStartingPoint(stStartingPoints(ix).Dh,
vRangeDH);
        stStartingPoints(ix).v50 = bumpStartingPoint(stStartingPoints(ix).v50,
vRangeV50);
    end
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

    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.MCV};
```

图 15 续

```
    if (bDB)
        sODBC = 'microcytosis';
        r = [];
        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', 'HDW', ...
        'muAge', 'alphaV', '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.r, 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(6)*stParameters.muAge/stParameters.mch^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.r, 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
```

图 15 续

```
    if (bDB)
        fastinsert(conn, 'ModelParameters', colNames, newData);
        close(conn);
    else
        newDataStart{1} = datenum(newDataStart(1));
        newData{4} = datenum(newData(4));
        t = cell2mat([newDataStart, newData]);
        sTimestamp = [datestr(now, 'yyyymmdd') '.' datestr(now,'HHMMSS')];
        dlmwrite([sAccession '.' sTimestamp '.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 = flipud(H);
    stParameters.V = V;
    stParameters.H = H;

    vEmpirical = strcmp({stSimulationParameters(:).sInitialDistribution}',
'Empirical');
    bEmpirical = all(vEmpirical);
    if (bEmpirical)
        stParameters.nMeanMarrowVolume = stSteadyState.rMCV;
        stParameters.nCVMarrowVolume = stSteadyState.rRDW;
        stParameters.nMeanMarrowHb = stSteadyState.rMCH;
        stParameters.nCVMarrowHb = stSteadyState.rHDW;
        stParameters.vStar = stSteadyState.MCV;
        stParameters.hStar = stSteadyState.MCH;
    else
        stParameters.nMeanMarrowVolume = stSteadyState.stAdviaFile.rMCV;
        stParameters.nCVMarrowVolume = stSteadyState.stAdviaFile.rRDW;
        stParameters.nMeanMarrowHb = stSteadyState.stAdviaFile.rMCH;
        stParameters.nCVMarrowHb = stSteadyState.stAdviaFile.rHDW;
        stParameters.vStar = stSteadyState.stAdviaFile.MCV;
        stParameters.hStar = stSteadyState.stAdviaFile.MCH;
    end

    stParameters.P_0 = stSteadyState.P_0;

    stParameters.nMarrowRho = stSteadyState.rRho;
    stParameters.muAgeTarg = 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;
```

图 15 续

```
    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.oQuasiLinearFit = getQuasiLinearFit(stParameters,
stSteadyState.vData, stSteadyState.hcData);
    stParameters.bPenalizeFalsePositives =
stSimulationParameters.bPenalizeFalsePositives;

    [x, resnorm, output, jacobian, stParameters] = optFDPDF(stSteadyState.vData,
stSteadyState.hcData, stParameters);
    stParameters.x = x;
    stParameters.output = output;
    stParameters.resnorm = resnorm;

    % Store 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.
        %%%
        %%%%%
        bLSQ = 0;
        bFMinSearch = 0;
        bPatternSearch = 1;
        bBounded = 1;

        sStartingPointMethod = '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 = 'MVN';
        sInitialDistribution = 'Empirical';
```

图 15 续

```
        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);
```

图 15 续

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


function stSimulationParameters = querySimulationParameters(sDate)
    sODBC = 'microcytosis';
    r = [];
    conn = database(sODBC, '', '');

    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))
            tSimulationParameters.ID = csSimulationParameters{ixStart, 1};
            tSimulationParameters.nDate = nDate;
            tSimulationParameters.sOptimizer = csSimulationParameters{ixStart,
3};
            tSimulationParameters.nMeshWidth = csSimulationParameters{ixStart,
5};
            tSimulationParameters.nMinV = csSimulationParameters{ixStart, 6};
            tSimulationParameters.nMaxV = csSimulationParameters{ixStart, 7};
            tSimulationParameters.nMinH = csSimulationParameters{ixStart, 8};
            tSimulationParameters.nMaxH = csSimulationParameters{ixStart, 9};
            tSimulationParameters.sStartPointMethod =
csSimulationParameters{ixStart, 10};
            tSimulationParameters.sFunctionalForm =
csSimulationParameters{ixStart, 11};
            tSimulationParameters.sInitialDistribution =
csSimulationParameters{ixStart, 12};
            tSimulationParameters.nMinAlpha = csSimulationParameters{ixStart,
13};
            tSimulationParameters.nMaxAlpha = csSimulationParameters{ixStart,
14};
            tSimulationParameters.sDeathFunction =
csSimulationParameters{ixStart, 15};
            tSimulationParameters.sOptimalTrend = csSimulationParameters{ixStart,
16};
            tSimulationParameters.nSteadyStateVersion =
csSimulationParameters{ixStart, 17};
            if (isnan(tSimulationParameters.nSteadyStateVersion ))
                tSimulationParameters.nSteadyStateVersion = 0;
            end
            tSimulationParameters.bPenalizeFalsePositives =
csSimulationParameters{ixStart, 18};
            if (isnan(tSimulationParameters.bPenalizeFalsePositives))
                tSimulationParameters.bPenalizeFalsePositives = 0;
```

图 15 续

```
            end
            stSimulationParameters = [stSimulationParameters;
tSimulationParameters];
        end
    end
    close(curs);
    close(conn);
end

% Get the starting points entered after the threshold date.
function stStartingPoints = queryStartingPoints(sDate)
    sODBC = 'microcytosis';
    r = [];
    conn = database(sODBC, '', '');

    curs = exec(conn, ['select * from StartingPoint']);
    curs = fetch(curs);
    csStartingPoint = curs.Data;
    stStartingPoints = [];
    for ixStart = 1:size(csStartingPoint, 1)
        nDate = datenum(csStartingPoint{ixStart, 2});
        if (nDate > datenum(sDate))
            tStartingPoint.ID = csStartingPoint{ixStart, 1};
            tStartingPoint.nDate = nDate;
            tStartingPoint.alpha = csStartingPoint{ixStart, 3};
            tStartingPoint.betav = csStartingPoint{ixStart, 4};
            tStartingPoint.betah = csStartingPoint{ixStart, 5};
            tStartingPoint.Dv = 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 smoothing of the sample.
    vi = linspace(min(vData), max(vData), round(range(vData)));
    hi = linspace(min(hData), max(hData), round(range(hData)));

    ctrs{1} = hi;
    ctrs{2} = vi;

    % Calculate the empirical distribution.
    P = hist3([hData, vData], ctrs);
    P = flipud(P);
    hi = fliplr(hi);
    vi = repmat(vi(:)', size(P, 1), 1);
    hi = repmat(hi(:), 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);
```

图 15 续

```
    k(ceil(n/2),ceil(n/2)) = nCenterWeight;
    Phat = conv2(P, k, 'same');
    Phat = sum(P(:))*Phat/sum(Phat(:));
    %ixCells = find(P > 5);
    %Phat = sum(P(ixCells))*Phat/sum(Phat(ixCells));
    %chi2 = sum((P(ixCells) - Phat(ixCells)).^2./P(ixCells));
end

function xi = getNormSpace(x)
    mu = median(x);
    sigma = iqr(x)*0.7314;
    normx = linspace((min(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
```

图 15 续

```
function stSteadyState = calculateSteadyState(conn, stAdviaFile, nVersion)
    stSteadyState = [];

    % Calculate the moments of the RBC distribution.
    [nMCV, nRDW, nCHCM, nHDW, nRho, v, h, nVersion] = getRBCPDF(stAdviaFile,
nVersion);

    % Calculate the moments of the reticulocyte distribution.
    [f, nrMCV, nrRDW, nrCHCM, nrHDW, nrRho, vr, hr, nVersion] =
getReticPDF(stAdviaFile, nVersion);

    nDate = now;
    sDate = datestr(nDate, 'mm/dd/yyyy HH:MM:SS PM');
    colNames = {'DateEntered', 'AdviaFile', 'MCV', 'RDW', 'MCH', 'HDW', 'rho',
...
        'rFraction', 'rMCV', 'rRDW', 'rMCH', 'rHDW', 'rRho', 'nVersion'};
    newData = {sDate, stAdviaFile.ID, nMCV, nRDW/100, nCHCM, nHDW/100, nRho(1,2),
...
        f, nrMCV, nrRDW/100, nrCHCM, nrHDW/100, nrRho(1,2), nVersion};
    fastinsert(conn, 'SteadyStateParameters', colNames, 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 caculations 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 > datenum(sDate));
        [t, ix] = sort([stSteadyState(ixCurrent).nDate]);
        stSteadyState = stSteadyState(ixCurrent(ix));
        stSteadyState = stSteadyState(end);
    end
end


function [nMCV, nRDW, nMCH, nHDW, nRho, v, hc, nVersion] = getRBCPDF(stAdviaFile,
nVersion)
    if (nargin < 2)
        nVersion = 0;
    end

    if (isstruct(stAdviaFile))
        sFCS = stAdviaFile.RBCFCSFile;
    else
        sFCS = stAdviaFile;
    end
```

图 15 续

```
    if (~isempty(sFCS) && ~strcmp(sFCS, 'null'))
        % Old version using FCS files.
        [nMCV, nRDW, nMCH, nHDW, nRho, v, hc, nVersion] =
compareMieTransforms(sFCS, nVersion);
    else
        % Assume it's an Abbott-supplied file.
        [nMCV, nRDW, nMCH, nHDW, nRho, v, hc] = getCSVPDF(stAdviaFile);
    end
end

function [nMCV, nRDW, nMCH, nHDW, nRho, 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);
    nRDW = 100*std(v)/nMCV;
    nHDW = 100*std(h)/nMCH;
    nRho = corrcoef([v h]);
end

function [f, nrMCV, nrRDW, nrCHCM, nrHDW, nrRho, 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, nrHDW, nrRho, vr, hcr, nVersion] =
estimateReticFraction(sFCS, nVersion);
    else
        % Assume it's an Abbott-supplied file.
        [f, nrMCV, nrRDW, nrCHCM, nrHDW, nrRho, vr, hcr] =
getCSVReticPDF(stAdviaFile);
    end
end

function [f, nrMCV, nrRDW, nrCHCM, nrHDW, nrRho, 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);
```

图 15 续

```
    nrRDW = 100*std(vr)/nrMCV;
    nrHDW = 100*std(hr)/nrCHCM;
    nrRho = corrcoef([vr hr]);
    f = 100*sum(ixRetic)/sum(ixRBC);
end


function [f, nMCV, nRDW, nMCH, nHDW, nRho, 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\jhiggins\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);
        nRDW = 100*std(v)/mean(v);
        nMCH = mean(v.*h/100);
        nHDW = 100*std(100*v.*h)/mean(100*v.*h);
        nRho = corrcoef([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))));
```

图 15 续

```
        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)));

        v = vFull(ixAnalyzed(ixGated(ixRetic)));
        h = hFull(ixAnalyzed(ixGated(ixRetic)));
        f = 100*length(ixRetic)/length(ixGated);
        nMCV = mean(v);
        nRDW = 100*iqr(v)*0.7413/nMCV;
        nMCH = mean(v.*h/100);
        nHDW = 100*iqr(100*v.*h)*0.7413/mean(100*v.*h);
        nRho = corrcoef([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(zBins(stPBANRES.nMinRTThreshold + [-1 0]))));
        ixLo = (find(d(ixAnalyzed(ixGated), 3) >=
zBins(stPBANRES.nLoThreshold)));
        ixMed = (find(d(ixAnalyzed(ixGated), 3) >=
zBins(stPBANRES.nMedThreshold)));

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

图 15 续

```
        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);
        nRDW = 100*iqr(v)*0.7413/nMCV;
        nMCH = mean(v.*h/100);
        nHDW = 100*iqr(100*v.*h)*0.7413/mean(100*v.*h);
        nRho = corrcoef([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 = prctile(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)));
        nMCV = mean(v);
        nMCH = mean(v.*h/100);
        v = v*stPBANRES.nMCVr/nMCV;
        h = h*(stPBANRES.nMCHr/stPBANRES.nMCVr)/(nMCH/nMCV);
        nMCV = mean(v);
        nMCH = mean(v.*h/100);
        f = 100*length(ixRetic)/length(ixGated);
        nRDW = 100*iqr(v)*0.7413/nMCV;
        nHDW = 100*iqr(100*v.*h)*0.7413/mean(100*v.*h);
        nRho = corrcoef([v v.*h]);
    elseif (nVersion == 5)
        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))));
```

图 15 续

```
        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 = prctile(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);
        nMCH = 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.nRDWr*stPBANRES.nMCVr/100) + stPBANRES.nMCVr;
        h_pg = (h_pg - nMCH)/nsdH_PG;
        h_pg = (h_pg*stPBANRES.nCHDWr) + stPBANRES.nMCHr;
        h = 100*h_pg./v;
        nMCV = mean(v);
        nMCH = mean(v.*h/100);

        nRDW = 100*iqr(v)*0.7413/nMCV;
        nHDW = 100*iqr(100*v.*h)*0.7413/mean(100*v.*h);
        nRho = corrcoef([v v.*h]);

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

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


function [stParameters, nSampleID] = getRBCPDFParameters(nModelParametersID)
```

图 15 续

```
    nSampleID = [];
    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.bMatchParameters = 0;
        stParameters.bPenalizeFalsePositives = 0;
        stParameters.nMinAlpha = NaN;
        stParameters.nMaxAlpha = NaN;


        stParameters.nMeanMarrowVolume = 109.3;
        stParameters.nCVMarrowVolume = 0.1129;
        stParameters.nMeanMarrowHb = 32.6;
        stParameters.nCVMarrowHb = 0.101;
        stParameters.nMarrowRho = 0.34;
        stParameters.vStar = 92.04;
        stParameters.hStar = 30.04;
        stParameters.nDays = 0;
        stParameters.muAgeTarg = 50;

        sdV = (stParameters.nCVMarrowVolume*stParameters.nMeanMarrowVolume);
        sdH = (stParameters.nCVMarrowHb*stParameters.nMeanMarrowHb);
        nCovar = stParameters.nMarrowRho*sdV*sdH;
        mSigma = [sdV^2 nCovar; nCovar sdH^2];
        P_0 = mvnpdf([V(:) H(:)], [stParameters.nMeanMarrowVolume
stParameters.nMeanMarrowHb], mSigma);
        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:
[0 120?];
        stParameters.betah = 9.0; % Rate of movement toward the "optimal" ratio:
[0 120?].
        stParameters.v50 = 77.9; %0.9*stParameters.vStar; % Volume where
P(clearance) = 50%: [0, 0.9]*MCV.
        stParameters.Dv = 2.6; %3.2774: [0 100?];
        stParameters.Dh = 0.0015; %.0793: [0 100?];
        stParameters.nSlope = 1;
        stParameters.alphaDecay = NaN;
        stParameters.sOptimalTrend = 'Linear';
```

图 15 续

```
    else
        sODBC = 'microcytosis';
        r = [];
        conn = database(sODBC, '', '');
        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 = expandFCSFilenames(csAdviaFile);
        nSampleID = afID;

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

        stParameters.nMeshWidth = csSimulationParameters{5};
        stParameters.nMaxVolume = csSimulationParameters{7};
        stParameters.nMinVolume = 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};
```

图 15 续

```
        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: [0 120?];
        stParameters.betah = csMP(15); % Rate of movement toward the "optimal"
ratio: [0 120?].
        stParameters.v50 = csMP(16); %0.9*stParameters.vStar; % Volume where
P(clearance) = 50%: [0, 0.9]*MCV.
        stParameters.Dv = csMP(17); %3.2774: [0 100?];
        stParameters.Dh = csMP(18); %.0793: [0 100?];
        if (size(csMP, 2) == 26)
            stParameters.nSlope = 1;
            stParameters.alphaDecay = NaN;
        elseif (size(csMP, 2) == 27)
            stParameters.nSlope = csMP(27);
            stParameters.alphaDecay = NaN;
        else
            stParameters.nSlope = csMP(27);
            stParameters.alphaDecay = csMP(28);
        end
        if (isnan(stParameters.nSlope))
            stParameters.nSlope = 1;
        end

        if (isnan(stParameters.alphaDecay))
            stParameters.alphaDecay = 5;
        end

        [f, nMCV, nRDW, nMCH, nHDW, nRho, vData, hcData] =
estimateReticFraction(csAdviaFile(6), ssVersion);
        [Phat, P, vi, hi] = makeEmpiricalPDF(vData, hcData);
        P_0 = interp2(vi, hi, Phat, V, H, 'linear', 0);
        P_0 = P_0(:)/sum(P_0(:));
        stParameters.P_0 = P_0;

        [nMCV, nRDW, nCHCM, nHDW, nRho, vData, hcData] =
compareMieTransforms(csAdviaFile(5), ssVersion);
        stParameters.vData = vData;
        stParameters.hcData = hcData;

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

        if 0
            sdV = (stParameters.nCVMarrowVolume*stParameters.nMeanMarrowVolume);
            sdH = (stParameters.nCVMarrowHb*stParameters.nMeanMarrowHb);
            nCovar = stParameters.nMarrowRho*sdV*sdH;
            mSigma = [sdV^2 nCovar; nCovar sdH^2];
            P_01 = mvnpdf([V(:) H(:)], [stParameters.nMeanMarrowVolume
stParameters.nMeanMarrowHb], mSigma);
```

图 15 续

```
            P_01 = reshape(sum(P_0(:))*P_01/sum(P_01(:)), size(V));
        end
    end
end

function [r, mu_v, cv_v, mu_h, cv_h] = computeRBCCorr(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.*repmat(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(repmat(mBirthAlive(:, end), [1 size(P, 2)]));
    Pnorm = P./repmat(sum(P')', [1 size(P, 2)]);
    Pf = Pnorm.*mWeights;
    if (size(Pf, 1) > 1)
        Pf = sum(Pf);
```

图 15 续

```
    end

    Pt = NaN*ones(nDays, size(Pf, 2));
    if (nargout == 4)
        tSamples = 1;
        for ix = 1:tSamples:(nDays - 1)
            mWeights = flipud(repmat(mBirthAlive(1:ix, ix), [1 size(P, 2)]));
            Ptemp = Pnorm(1:ix, :).*mWeights;
            if (size(Ptemp, 1) > 1)
                Ptemp = sum(Ptemp);
            end
            Pt(max(1, floor(ix/tSamples)), :) = Ptemp;
        end
        Pt(end, :) = Pf;
    end
end

function [P, t, V, H, stParameters] = fdRBC(stParameters)
    P =[];
    t = [];
    % Get the parameters.
    if (~exist('stParameters'))
        stParameters = getRBCPDFParameters();
    end
    vt = [0:stParameters.nDays];
    V = stParameters.V;
    H = stParameters.H;
    P_0 = stParameters.P_0;

    % Setup the clearance function.
    pCleared = getCleared(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, 1) + 1):size(V, 1):(numel(V) - 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);
    mHDot = hdot(stParameters, V, H);

    if (0)
        % Code to draw the velocity fields.
        clf
        nRange = 0.25;
        [V,H] = meshgrid([stParameters.nMinVolume:.3:stParameters.nMaxVolume],
...
            [stParameters.nMinHb:0.1:stParameters.nMaxHb]);
        mVDot = vdot(stParameters, V, H);
        mHDot = hdot(stParameters, V, H);
        VC = H./V;
```

图 15 续

```
        ix = find(VC(:) > (1.0 -
nRange)*(stParameters.hStar/stParameters.vStar));
        ix = intersect(ix, find(VC(:) < (1.0 +
nRange)*(stParameters.hStar/stParameters.vStar)));
        mLength = mVDot.^2 + mHDot.^2;
        nScale = 4;
        %pvf = quiver(V(ix), H(ix),  -1*(abs(mVDot(ix)).^nScale), -
1*(abs(mHDot(ix)).^nScale),'autoscalefactor',1, 'linewidth', 2, 'maxheadsize',
0.5);
        plot([0 2*stParameters.vStar],[0 2*stParameters.hStar], 'color', 'k',
'linewidth', 4);
        hold on;
        pvf = quiver(V(ix), H(ix),  -1*(abs(mVDot(ix))/nScale), -
1*(abs(mHDot(ix))/nScale),'autoscale', 'off', 'linewidth', 2, 'maxheadsize', 0.2,
'color', 'b');
        yl = [29.5 30.5];
        %yl = [27.5 32.5];
        xl = (stParameters.vStar/stParameters.hStar)*yl;
        set(gca,'ylim', yl);
        set(gca,'xlim', xl);
        nSize = 20;
        xlabel('volume (fL)', 'fontsize', nSize);
        ylabel('hemoglobin (pg)', 'fontsize', nSize);
        set(gca,'fontsize',nSize);
    end

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

    [spLaplacian, L] = getLaplacian(stParameters, numel(mVDot), ixTopRight,
ixBottomRight, ixBottomLeft, ixTopLeft, ...
        ixRightEdge, ixTopEdge, ixLeftEdge, ixBottomEdge, ixInterior, ixVStep);
    spJacobian = spones(J + L);
    stParameters.L = L;

    if (stParameters.nDays)
        odeOptions = odeset('Vectorized', 'on', 'JPattern', spJacobian,
'OutputFcn', 'odeplot');
        odeOptions = odeset('Vectorized', 'on', 'JPattern', spJacobian, 'AbsTol',
1e-6, 'OutputFcn', 'odeplot');
        odeOptions = odeset('Vectorized', 'on', 'JPattern', spJacobian);
        optset = optimset('TolX', 10^(-12), 'MaxFun', 1e10, 'MaxIter', 1e10);
        [t, P] = ode15s(@fUpwind, vt, P_0, odeOptions);
        %[t, P] = fsolve(@fUpwind, P_0, optset);
        %[t, P] = fminsearch(@fUpwind, P_0, optset);
    end

    function dP_dt = fUpwind(t, P)
        dP_dt = zeros(size(P,1), size(P, 2));

        [dP_Cleared, nCleared] = clearRBC(pCleared, P, V, H);
        P_born = P_0*nCleared;
        if 0
            dP_dt = J*P  + L*P - dP_Cleared + P_born;
        else
            dP_dt = J*P + L*P - dP_Cleared;
        end
    end
end
```

图 15 续

```
function pCleared = getCleared(stParameters, V, H)
    nSlope = stParameters.nSlope;
    if (isfield(stParameters, 'sDeathFunction'))
        assert(strcmp(stParameters.sDeathFunction, 'Projection') ||
strcmp(stParameters.sDeathFunction, 'Dimensionless Projection') || ...
            strcmp(stParameters.sDeathFunction, 'Tunable Dimensionless
Projection') || strcmp(stParameters.sDeathFunction, '0.1 Dimensionless
Projection'));

        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.oQuasiLinearFit.a);
            mAngle = atan((H - stParameters.oQuasiLinearFit.b)./V);
            mNorm = (V.^2 + (H - stParameters.oQuasiLinearFit.b).^2).^(1/2);
            vr50 = (stParameters.v50^2 +
(stParameters.v50*stParameters.oQuasiLinearFit.a)^2)^(1/2);
        else
            theta = atan(stParameters.hStar/stParameters.vStar);
            mAngle = atan(H./V);
            mNorm = (V.^2 + H.^2).^(1/2);
            vr50 = (stParameters.v50^2 +
(stParameters.v50*stParameters.hStar/stParameters.vStar)^2)^(1/2);
        end
        Vr = mNorm.*cos(theta - mAngle);

        if (strcmp(stParameters.sFunctionalForm, 'Hybrid') || ...
                strcmp(stParameters.sFunctionalForm, 'Exponential') || ...
                strcmp(stParameters.sFunctionalForm, 'Hybrid Exponential') || ...
                strcmp(stParameters.sFunctionalForm, 'Decaying Additive
Exponential') || ...
                strcmp(stParameters.sFunctionalForm, 'Additive Exponential'))
            if (strcmp(stParameters.sDeathFunction, 'Dimensionless Projection')
|| ...
                    strcmp(stParameters.sDeathFunction, 'Tunable Dimensionless
Projection') || ...
                    strcmp(stParameters.sDeathFunction, '0.1 Dimensionless
Projection'))
                pCleared = 1./(1 + exp(nSlope*100*(Vr - vr50)/vr50));
            else
                pCleared = 1./(1 + exp(nSlope*(Vr - vr50)));
            end
        elseif (strcmp(stParameters.sFunctionalForm, 'Simple'))
            pCleared = (Vr < vr50);
        end
        pCleared = reshape(pCleared, size(V));
    else
        theta = atan(stParameters.hStar/stParameters.vStar);
        R = [cos(theta) sin(theta); -1*sin(theta) cos(theta)];
        Vr = R(1,:)*[V(:)'; H(:)'];
        vr50 = R(1,:)*[stParameters.v50;
stParameters.v50*stParameters.hStar/stParameters.vStar];
        if (strcmp(stParameters.sFunctionalForm, 'Hybrid') ||
strcmp(stParameters.sFunctionalForm, 'Exponential') || ...
                strcmp(stParameters.sFunctionalForm, 'Hybrid Exponential') ||
strcmp(stParameters.sFunctionalForm, 'Additive Exponential') || ...
```

图 15 续

```
                strcmp(stParameters.sFunctionalForm, 'Decaying Additive
Exponential'))
            pCleared = 1./(1 + exp(nSlope*100*(Vr - vr50)/vr50));
        elseif (strcmp(stParameters.sFunctionalForm, 'Simple'))
            pCleared = (Vr < vr50);
        end
        pCleared = reshape(pCleared, size(V));
    end
end

function [dP_Cleared, nCleared] = clearRBC(pCleared, P, V, H)
    % The probability of being cleared is sigmoidal (logistic).
    pCleared = repmat(pCleared(:), 1, size(P, 2));

    % Keep track of how many (probability density) are cleared.
    dP_Cleared = pCleared.*P;
    nCleared = sum(dP_Cleared);
end

function [spLaplacian, L] = getLaplacian(stParameters, nVar, ixTopRight,
ixBottomRight, ixBottomLeft, ixTopLeft, ...
        ixRightEdge, ixTopEdge, ixLeftEdge, ixBottomEdge, ixInterior, ixVStep)

    Lv = -2*speye(nVar);
    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(ixTopEdge)
        Lv(ixTopEdge(ix), [(ixTopEdge(ix) - ixVStep) (ixTopEdge(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) (ixBottomEdge(ix) +
ixVStep)]) = 1;
    end
    for ix = 1:length(ixInterior)
        Lv(ixInterior(ix), [(ixInterior(ix) - ixVStep) (ixInterior(ix) +
ixVStep)]) = 1;
    end
    Lv = stParameters.Dv*Lv/(stParameters.nMeshWidth^2);

    Lh = -2*speye(nVar);
    Lh(ixTopRight, [(ixTopRight + 1)]) = 1;
    Lh(ixBottomRight, [(ixBottomRight - 1)]) = 1;
    Lh(ixBottomLeft, [(ixBottomLeft - 1)]) = 1;
    Lh(ixTopLeft, [(ixTopLeft + 1)]) = 1;

    for ix = 1:length(ixRightEdge)
        Lh(ixRightEdge(ix), [(ixRightEdge(ix) + 1) (ixRightEdge(ix) - 1)]) = 1;
    end
    for ix = 1:length(ixTopEdge)
        Lh(ixTopEdge(ix), [(ixTopEdge(ix) + 1)]) = 1;
```

图 15 续

```
    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(mVDot));

    %%% 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;
```

图 15 续

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

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

    spJacobian = spones(J);
end

function dv_dt = vdot(stParameters, V, H)
    dV = getVDistance(stParameters, V, H);
    if (strcmp(stParameters.sFunctionalForm, 'Hybrid'))
        dv_dt = -1*stParameters.alphav*stParameters.vStar*...
            exp(stParameters.betav*(dV));
    elseif (strcmp(stParameters.sFunctionalForm, 'Exponential'))
        dv_dt = -1*(stParameters.alphav*V).*...
            max(1, stParameters.betav*(dV));
    elseif (strcmp(stParameters.sFunctionalForm, 'Piecewise'))
        deltaV = max(0, dV);
        nFast = min(zeros(size(deltaV)), deltaV -
stParameters.nCVMarrowVolume)*stParameters.betav;
        nSlow = max(deltaV, stParameters.nCVMarrowVolume)*stParameters.betav/4;
        dv_dt = -1*stParameters.alphav*(stParameters.vStar/stParameters.hStar)*(1
+ nFast + nSlow);
    elseif (strcmp(stParameters.sFunctionalForm, 'Simple'))
        dv_dt = -1*stParameters.alphav*stParameters.vStar*...
            max(1, stParameters.betav*(dV));
    elseif (strcmp(stParameters.sFunctionalForm, 'Additive Exponential'))
        dv_dt = -1*stParameters.alphav*V - stParameters.betav*(max(0, dV));
    elseif (strcmp(stParameters.sFunctionalForm, 'Decaying Additive
Exponential'))
        mNorm = (V.^2 + H.^2).^(1/2);
        starNorm = sqrt(stParameters.vStar^2 + stParameters.hStar^2);
        dv_dt = -1*stParameters.alphav*V.*exp(stParameters.alphaDecay*(mNorm -
starNorm)/starNorm) - stParameters.betav*(max(0, dV));
    elseif (strcmp(stParameters.sFunctionalForm, 'Hybrid Exponential'))
        dv_dt = -1*stParameters.alphav*V.*...
            exp(stParameters.betav*(dV));
    end
end

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

function dh_dt = hdot(stParameters, V, H)
    dH = getHDistance(stParameters, V, H);
    if (strcmp(stParameters.sFunctionalForm, 'Hybrid'))
        dh_dt = -1*stParameters.alphah*stParameters.hStar*...
            exp(stParameters.betah*(dH));
    elseif (strcmp(stParameters.sFunctionalForm, 'Exponential'))
        dh_dt = -1*(stParameters.alphah*H).*...
            max(1, stParameters.betah*(dH));
    elseif (strcmp(stParameters.sFunctionalForm, 'Piecewise'))
```

图 15 续

```
        deltaH = max(0, dH);
        nFast = min(zeros(size(deltaH)), deltaH -
stParameters.nCVMarrowHb)*stParameters.betah;
        nSlow = max(deltaH, stParameters.nCVMarrowHb)*stParameters.betah/4;
        dh_dt = -1*stParameters.alphah*(stParameters.vStar/stParameters.hStar)*(1
+ nFast + nSlow);
    elseif (strcmp(stParameters.sFunctionalForm, 'Simple'))
        dh_dt = -1*stParameters.alphah*stParameters.hStar*...
            max(1, stParameters.betah*(dH));
    elseif (strcmp(stParameters.sFunctionalForm, 'Additive Exponential'))
        dh_dt = -1*stParameters.alphah*H - stParameters.betah*(max(0, dH));
    elseif (strcmp(stParameters.sFunctionalForm, 'Decaying Additive
Exponential'))
        mNorm = (V.^2 + H.^2).^(1/2);
        starNorm = sqrt(stParameters.vStar^2 + stParameters.hStar^2);
        dh_dt = -1*stParameters.alphah*H.*exp(stParameters.alphaDecay*(mNorm -
starNorm)/starNorm) - stParameters.betah*(max(0, dH));
    elseif (strcmp(stParameters.sFunctionalForm, 'Hybrid Exponential'))
        dh_dt = -1*stParameters.alphah*H.*...
            exp(stParameters.betah*(dH));
    end
end

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

function [x, resnorm, output, jacobian, stParameters] = optFDPDF(vData, hcData,
stParameters)
    if (nargin == 2)
        stParameters = getRBCPDFParameters();
    end

    bMatchParameters = stParameters.bMatchParameters;
    bPenalizeFalsePositives = stParameters.bPenalizeFalsePositives;

    % Use an empirical target distribution.
    [Phat, P_empirical, vi, hi] = makeEmpiricalPDF(vData, hcData);
    ixCells = find(Phat > 5);
    nCells = sum(Phat(ixCells));
    nCells = sum(Phat(:));
    stParameters.df = numel(ixCells);
    [r_targ, mu_v_targ, rdw_targ, mu_h_targ, hdw_targ] = computeRBCCorr(Phat(:)',
vi, hi);
    mu_age_targ = stParameters.muAgeTarg;
    vTarget = [r_targ , ...
        mu_v_targ, ...
        rdw_targ, ...
        mu_h_targ, ...
        hdw_targ, ...
        mu_age_targ];

    if (strcmp(stParameters.sOptimalTrend, 'Quasi-linear'))
        x0 = [stParameters.alphav, ...
            stParameters.betav, ...
```

图 15 续

```
            stParameters.betah, ...
            stParameters.v50, ...
            stParameters.Dv, ...
            stParameters.Dh, ...
            stParameters.alphah];
    else
        x0 = [stParameters.alphav, ...
            stParameters.betav, ...
            stParameters.betah, ...
            stParameters.v50, ...
            stParameters.Dv, ...
            stParameters.Dh];
    end
    if (strcmp(stParameters.sDeathFunction, 'Tunable Dimensionless Projection'))
        x0 = [x0, stParameters.nSlope];
    end
    if (strcmp(stParameters.sFunctionalForm, 'Decaying Additive Exponential'))
        x0 = [x0, stParameters.alphaDecay];
    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);
        %optset = optimset('Display', 'iter');
        [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.alphav = x(1); % Rate of translation down the "optimal" HC
ratio.
        bQuasiLinear = strcmp(stParameters.sOptimalTrend, 'Quasi-linear');
        if (bQuasiLinear)
            stParameters.alphah = x(7); % Rate of translation down the "optimal"
HC ratio.
        else
```

图 15 续

```
            stParameters.alphah = x(1); % Rate of translation down the "optimal"
HC ratio.
        end
        stParameters.betav = x(2); % Rate of movement toward the "optimal" ratio.
        stParameters.betah = x(3); % Rate of movement toward the "optimal" ratio.
        stParameters.v50 = x(4); % Volume where P(clearance) = 50%.
        stParameters.Dv = x(5);
        stParameters.Dh = x(6);

        bTunable = strcmp(stParameters.sDeathFunction, 'Tunable Dimensionless
Projection');
        bDecaying = strcmp(stParameters.sFunctionalForm, 'Decaying Additive
Exponential');

        if (bTunable)
            stParameters.nSlope = x(7 + bQuasiLinear);
        end
        if (bDecaying)
            stParameters.alphaDecay = x(7 + bQuasiLinear + bTunable);
        end

        % Simulate the evolution to steady state.
        [P, t, V, H, tParameters] = fdRBC(stParameters);
        if (stParameters.nDays)
            % Calculate the steady state parameters.
            [mBirthAlive, vMeanAge, Pf] = computeAgeDistribution(P);
            muAge = vMeanAge(end);
        else
            [Pf, A, muAge] = ssRBC(tParameters.J, tParameters.L, tParameters.P_0,
tParameters.pCleared);
            Pf = Pf(:)';
        end
        [r, mu_v, cv_v, mu_h, cv_h] = computeRBCCorr(Pf, V, H);
        stParameters.r = r;
        stParameters.mcv = mu_v;
        stParameters.mch = mu_h;
        stParameters.rdw = cv_v;
        stParameters.hdw = cv_h;
        stParameters.muAge = muAge;


        vEstimate = [r, ...
            mu_v, ...
            cv_v, ...
            mu_h, ...
            cv_h, ...
            muAge];

        if (bMatchParameters)
            % Try to match the parameters of the empirical distribution.
            vResiduals = vTarget - vEstimate;
            y = 100*(1./vTarget).*vResiduals;

            % The average age is a weaker constraint.
            y(end) = y(end)/100;
        else
            % Evaluate the simulated distribution at the points in the
            % empirical distribution.
            Pf_hat = interp2(V, H, reshape(Pf, size(V)), vi, hi, 'linear', 0);
            Pf_hat = nCells*Pf_hat/sum(Pf_hat(:));
            ixCellSimulated = find(Pf_hat > 5);
            ixCellSimulatedFP = setdiff(ixCellSimulated, ixCells);
```

图 15 续

```
            %nHat = sum(Pf_hat(ixCells));
            %Pf_hat = Pf_hat*(nCells/sum(Pf_hat(ixCells)));
            vResiduals = (Pf_hat(ixCells) - Phat(ixCells))./sqrt(Phat(ixCells));
            yFP = (Pf_hat(ixCellSimulatedFP) -
5)./sqrt(Pf_hat(ixCellSimulatedFP));
            y = [vResiduals(:); (numel(vResiduals)/10)*(mu_age_targ -
muAge)/sqrt(mu_age_targ)];
            if (bPenalizeFalsePositives)
                yFP = (Pf_hat(ixCellSimulatedFP) -
5)./sqrt(Pf_hat(ixCellSimulatedFP));
                y(end) = y(end) + norm(yFP)^2;
            end
            %norm(vResiduals)^2
            %[h, p, stats] = chi2gof(Pf_hat(ixCells), 'expected',
            %P_empirical(ixCells));
            %compare PDFs(Pf_hat, Phat, vi, hi);
        end
        disp(sprintf('df(%d); r: %0.2f (=> %0.2f); MCV: %0.2f (=> %0.2f); RDW:
%0.2f (=> %0.2f); MCH: %0.2f (=> %0.2f); HDW: %0.2f (=> %0.2f); muAge: %0.2f (=>
%0.2f)', ...
            numel(vResiduals), r, r_targ, mu_v, mu_v_targ, cv_v, rdw_targ, mu_h,
mu_h_targ, cv_h, hdw_targ, muAge, mu_age_targ));
        if (~strcmp(stParameters.sOptimizer, 'lsqnonlin'))
            y = norm(y)^2;
            if (strcmp(stParameters.sOptimizer, 'fminsearch'))
                lbPenalty = any(x < lb);
                ubPenalty = any(x > ub);
                y = y + 1e7*(lbPenalty + ubPenalty);
            end
        end
    end
end
```

图 15 续

```
function [Pf, A, muAge] = ssRBC(J, L, P_0, pCleared)
    % Make a sparse diagonal from pCleared.
    spD = sparse(numel(pCleared));
    for ix = 1:numel(pCleared)
        spD(ix,ix) = pCleared(ix);
    end

    A = (L + J - spD);
    Pf = -1*A\(P_0(:));
    nMin = min(Pf(:));
    if (nMin < -eps)
        Pf = Pf + abs(nMin);
    end
    muAge = sum(Pf)/2;
end
```

图 16

```
function [p, u] = calculateCompensation(sFCS, bPlot, nVersion, bRetic)
    if (nargin < 3)
        bRetic = 0;
        nVersion = 3;
        if (nargin < 2)
            bPlot = 0;
            if (nargin < 1)
                sFCS = getFCS();
            end
        end
    end

    % Load the FCS.
    p = [];
    for ix = 1:size(sFCS, 1)
        [nMCV, nRDW, nMCH, nHDW, nRho, v, hc, nVersion] =
compareMieTransforms(sFCS{ix}, nVersion);

        if (bRetic == 3)
            [f, nMCV, nRDW, nMCH, nHDW, nRho, v, hc, nVersion] =
estimateReticFraction(sFCS{ix}, nVersion);
        end
        % Get the empirical retics and steady state.
        [Phat, P, vi, hi] = makeEmpiricalPDF(v, hc);

        h = v.*hc/100;
        %vi = vi/(Phat(:)'*vi(:)/sum(Phat(:)));
        %hi = hi/(Phat(:)'*hi(:)/sum(Phat(:)));
        %v = v/mean(v);
        %h = h/mean(h);

        if (bPlot)
            %figure;
            if (bRetic == 2)
                cColor = 'r';
            else
                cColor = 'b';
            end
            plotContourFit(vi, hi, Phat, cColor, bRetic);
        end

        % Transform.
        bRotateAndScale = 1;
        if (bRotateAndScale)
            theta = -atan(nMCH/nMCV);
            u = v*cos(theta) - h*sin(theta);
            w = v*sin(theta) + h*sin(theta);

            ui = vi*cos(theta) - hi*sin(theta);
            wi = vi*sin(theta) + hi*cos(theta);
        else
            % Project, rotate, and scale;
            theta = atan(nMCH/nMCV);

            p1 = [cos(theta)^2; sin(theta)*cos(theta)];
            p2 = [sin(theta)*cos(theta); sin(theta)^2];
            P = [p1 p2];
```

图 17

```
            theta = -1*theta;
            r1 = [cos(theta); sin(theta)];
            r2 = [-1*sin(theta); cos(theta)];
            R = [r1 r2];
            u = R*P*[v'; h'];
            v = u(2,:)';
            u = u(1,:)';
        end

        tp = [sum([u u u u u u u u] < repmat([.70 0.75 0.79 0.82 0.85 0.90 0.95
1]*mean(u), numel(u), 1))/numel(u)];

        ixLow = find(u < median(u));
        nLowVar = sqrt(sum(((u(ixLow) - median(u)).^2)/numel(ixLow)));
        tp = [tp nLowVar];
        p = [p; tp];
    end
end

function sFCS = getFCS()
    sODBC = 'microcytosis';
    r = [];
    conn = database(sODBC, '', '');

    %curs = exec(conn, ['select [RBC FCS File] from AdviaFile where [Diagnosis] =
''normalTrain'''']);
    curs = exec(conn, ['select [RBC FCS File] from AdviaFile where [Diagnosis] =
''ironTest'''']);
    curs = fetch(curs);
    sFCS = curs.Data;

    close(conn);

if 0
   figure
   ix = 8;
   subplot(211)
   hist(mNorm(:,ix),20)
   xl1 = get(gca,'xlim')
   subplot(212)
   hist(mIron(:,ix))
   xl2 = get(gca,'xlim')
   set(gca,'xlim',[min(xl1(1), xl2(1)) max(xl1(2), xl2(2))]);
   subplot(211);
   set(gca,'xlim',[min(xl1(1), xl2(1)) max(xl1(2), xl2(2))]);
end
end
```

图 17 续