

## Supporting Information

### 1. The Source Code of a short Algorithm compatible to the CELLmicrocosmos 2.2 MembraneEditor

```
// the main method for the lipid placement
// other methods provide options to create a initial GUI window, runtime options,
// supporting information for author and description, etc.
// the following variables has to be defined before: this could be done by the programmer
// of this algorithm or by the user if the programmer implemented a GUI for these values:
// - boolean rotateLipids
// - int desiredNumberOfLipids
public void fillWithLipids() throws Exception {
    int currentCount = 0;
    // sets the actual membrane side to the intracellular one
    int mSide = MembraneAlgorithm.INTRA;
    // repeats the placement process until the algorithm is stopped manually by the user
    // or the defined number of lipids in the membrane is reached
    while (stop == false && currentCount < this.desiredNumberOfLipids) {
        // changes each time the the actual membrane side to extra- or intra-cellular
        if (mSide == MembraneAlgorithm.INTRA)
            mSide = MembraneAlgorithm.EXTRA;
        else
            mSide = MembraneAlgorithm.INTRA;
        // creates a random position between the 0/0 point and the width (x) and
        // height (z/respectively length) of the membrane
        Point2f randomPosition = this.createRandomXZPosition(new Rectangle(0,
            0, this.getMembraneSize().width, this.getMembraneSize().height));
        Lipid newLipid;
        // checks if there is a microdomain at the computed random position
        int domainID = this.getMicroDomainAtPoint(randomPosition);
        // if there is a microdomain at this position,
        // the new lipid is created in this microdomain
        if (domainID > 0) {
            MicroDomain domain = this.getMicroDomainAt(domainID - 1);
            newLipid = mSide == EXTRA ? domain.createNextExtracellularLipid() :
                domain.createNextIntracellularLipid();
        }
        // if there is no microdomain, the lipid is added to the default area of the membrane
        } else {
            newLipid = mSide == EXTRA ? this.createNextExtracellularLipid()
                : this.createNextIntracellularLipid();
        }
        // cancel the adding of this lipid, if the creation of a new lipid did not succeed
        if (newLipid == null)
            continue;
        // define the position of the newly created lipid as the random position already created
        newLipid.setPosition(new Point3f(randomPosition.x,
            newLipid.getYTranslation(), randomPosition.y));
        // GUI: if the user chooses to alter the rotation, do it randomly here
        if (this.rotateLipids)
            newLipid.rotY(new Random().nextInt(359) + new Random().nextFloat());
        // if the new lipid intersects the membrane borders or another molecule,
        // the cancel the adding of this lipid
        if (this.intersectsBorders(newLipid) || this.intersects(newLipid))
            continue;
        // finally, the new lipid is added to the predefined domain
        if (mSide == EXTRA)
            this.addExtracellularLipid(newLipid);
        else
            this.addIntracellularLipid(newLipid);
        // iterate the number of lipids, needed to find out if the algorithm is finished
        currentCount++;
    }
}
```

## 2. PDB Files Information

**Table S1.** The lipid PDB files used in the application cases.

Abbreviation	Compound PDB ID	PDB source file	ideal/extracted atoms	lipid formula	Ref.
Chol	clr	1lri	28/28	C27 H46 O	6, 49
CL	cdn	2uwu	81/77	C58 H120 O17 P2	6, 49
PC	psc	1v54	52/52	C42 H81 N O8 P 1+	6, 49
PC, DLPC	dlp	1ln1	54/54	C44 H80 N O8 P	6, 49
PC, DPPC	-	DPPC.pdb	50/50	C40 H80 N O8 P	12
PE	pew	2hit	51/51	C39 H76 BR2 N O8 P	6, 49
PI	pib	1h6h	35/35	C17 H32 O16 P2	6, 49
PS	ps2	2hj6	54/54	C40 H76 BR2 N O10 P	6, 49
SM	-	sphingomyelin.pdb	50/50	C24 H49 N2 O6 PR	68

## 3. Table showing random-placing-based algorithms

This table shows all algorithms using random-placing-based methods based on section 3.1.1. The Random Placing Longterm values were not shown in Table 1.

**Table S2.** Random-placing-based algorithmic lipid compositions in outer mitochondrial membrane of rat hepatocyte<sup>(66)</sup> <sup>a</sup>

	experim. % <sup>(66)</sup>	CmME initial values (%)	Random Placing <sup>b</sup>	Random Placing Longterm <sup>c</sup>	Adv. Rand. Placing	Adv. Rand. Placing+ <sup>d</sup>	Distri- butor+ <sup>e</sup>	Distri- butor <sup>f</sup>
CL	9 ±2	9	8.805	8.92	8.886	8.929	8.9	8.805
PC	48 ±5	49.1	49.161	49.121	49.171	49.074	49.132	49.161
PE	31 ±1.6	31	31.027	31.03	31.043	30.952	31.042	31.027
PI	9.9 ±1.7	9.9	9.958	9.925	9.953	9.987	9.913	9.958
PS	1 ±0.3	1	1.048	1.005	0.984	1.058	1.013	1.048
deviation sum <sup>h</sup>			0.389	0.181	0.297	0.288	0.2	0.389
Number of Lipids			464/ 490	464/ 490	425/ 419	761/ 751	686/ 696	464/ 490
Average Lipid Density (Å <sup>2</sup> )			86.20/ 81.63	86.20/ 81.63	94.11/ 95.46	52.56/ 53.26	58.30/ 57.47	86.20/ 81.63
Compu- tation Time <sup>i</sup>			1m [60672]	40m 1s [2401765]	13s [13343]	39m 37s [2377344]	11m 32s [692750]	47s [47516]

Table S2: <sup>a</sup> See also Table 1/Figure 5. The CmME membrane compositions were tested on a 200x200 Å<sup>2</sup> membrane with standard algorithm values and atom collision only. The values represent the average of both layers. The seed for the random number generation was 22. Special algorithm settings were as follows: <sup>b</sup> stopped after 1 min, run with visualization <sup>c</sup> stopped after 40 min, run without visualization <sup>d</sup> with superb density distribution <sup>e</sup> the absolute lipid numbers were taken from the resulting membrane of the Simulated Annealing Algorithm <sup>f</sup> the absolute lipid numbers were set to 400 from the beginning <sup>g</sup> wander for fellow and dance option, 50 possible missteps, this algorithm is not included in CmME and can be downloaded separately <sup>h</sup> based on the CmME initial values <sup>i</sup> test configuration: Intel® Pentium® 4; 3 GHz (no HyperThreading); 2 GB Ram (Java Virtual Memory: 1 GB); Windows XP SP3; Java 1.6.0\_21; Java3D 1.5.2 fcs (build4); computation without visualization (except <sup>b</sup>)

#### 4. Supplementary PDB-Files

The Supplementary data contains PDB files discussed in the sections 3.1

The files are:

**rat\_liver\_mitochondria\_hovius1990\_hic-up\_rev.zip**

This PDB files contain reverse parsing information for CmME. After installing and starting CmME at

<http://Cm2.CELLmicrocosmos.org>

the user has to choose from the menu

File → Rebuild Membrane Model from PDB File

and select a PDB file containing reverse parsing information.

**rat\_liver\_mitochondria\_hovius1990\_hic-up.zip**

**Additional files** can be found at this website:

[http://www.cellmicrocosmos.org/index\\_cm2.php?inputPage=cm2project\\_supplement\\_JCIM](http://www.cellmicrocosmos.org/index_cm2.php?inputPage=cm2project_supplement_JCIM)

#### 5. Plug-in-algorithms

One of the main ideas behind CmME was the possibility of offering an interface to create algorithms without the need to know, understand or edit the original source code. All required methods are pre-implemented. Optionally a GUI window as well as a runtime settings frame can be created for user interactions. Lipids and proteins including their identifiers, positions, orientations, membrane-layers and

stacks, microdomains, single molecules and atoms can be accessed and recomputed. The algorithms based on Java can even be imported at runtime.

The source code of a very short exemplary algorithm as well as the algorithm complexities can be found in the Supporting Information. The documentation of the plug-in-interface as well as the optional algorithm called “The Wanderer”, providing very high lipid density, are available free-of-charge on our website.

## **6. PDB Properties**

To ensure compatibility between the PDB format and different PDB viewers, a PDB Properties dialog has been created. This is used to adjust the exported PDB format to the requirements of the visualization and simulation software. The settings affect, amongst others, the serial numbers and chain IDs important for the secondary structure, the coordinate transformation and the integration of reverse parsing information. This may be important if the exported membrane is altered with other software, GROMACS<sup>(16)</sup> for example, and should thereafter be re-imported for further editing.

In addition, a Title Editor has been added to enable the author to change settings such as the creation date, the title, source and remark section of the resulting PDB file.

## 7. Technicalities

Based on Java 6 and Java3D 1.5.2, CmME is Webstart ready. It has been tested on various Windows, Linux and Mac OS X configurations. This program can be directly installed and executed from its web resource:

<http://Cm2.CELLmicrocosmos.org>

Recommended system configuration: 2 GHz CPU, 1 GB Ram, 3D Graphic Card, 512 MB Java Heap Space (more heap space is needed for very large membranes). 3D Stereoscopy is implemented for nVidia® Quadro® FX cards and compatibles.

Includes Packages:

- Infonode Docking Windows (GNU General Public License/GPL)
- Jmol (GNU Lesser General Public License/LGPL)
- JFreeChart (GNU Lesser General Public License/LGPL)

CELLmicrocosmos 2.2 MembraneEditor (CmME) is released on GNU General Public License/GPL.

## 8. Visualization Statement

Figure 1 was created with Macromedia® Freehand®/ Adobe® Photoshop®. The visualizations in Table 3 and Figures 2, 3, 6-11 were visualized with CmME. Figure 7 was constructed using CmME and different layers in Adobe® Photoshop®. Figure 4 was created with OpenOffice. Figure 5 was exported from CmME to Grace. Figure 12, 14 and 15 were exported from GROMACS to Grace. Figure 13 is based on VMD visualizations.