Supplementary information 1
Video showing a tomogram of a purified frozen hydrated *S. aureus* sacculus.

(Seconds 0 to 4) The video “Video_SI1.mp4” shows the tomogram from the bottom to the top. There is a first stopping point which corresponds to where image ED6b was extracted from and then a second stopping point which corresponding to ED6a.

Each slice shown on the tomogram correspond to 1.3 nm. Images ED6a-d correspond to 10 slices thick, which means they represent 13 nm of the tomogram.

(Seconds 4 to 7) Later on in the video, the slices are shown from top to bottom, with an isosurface (in green) manually drawn showing the outline of the sacculus.

Supplementary information 2
MATLAB code for determining glycan strand orientation

```
% read the image into MATLAB and convert it to grayscale

I = imread(‘image_name.jpg'); %enter name of file within ' '
Igray = rgb2gray(I);
figure, imshow(I);


%  We will clean it up with a few morphological operations
Ibw = im2bw(Igray, graythresh(Igray));
se = strel('line',3,200);
cleanI = imdilate(~Ibw,se);
figure, imshow(cleanI);


% Perform a Hough Transform on the image The Hough Transform identifies
lines in an image
[H,theta,rho] = hough(cleanI);
P = houghpeaks(H,300,'threshold',ceil(0.5*max(H(:)))); %find peaks with a
threshold of minimum signal to be considered a peak


%the houghlines function defines how to draw the lines. Variable parameters
are: Fill Gap (defines minimum space between two lines). Min Length defines
the minimum size of a marked line. The numbers were optimized using the set
of images available.
lines = houghlines(cleanI,theta,rho,P,'FillGap',3,'MinLength',15);


%Plot the lines on top of image marking the beginning and end
hold on
max_len = 0;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green','LineStyle','-');


   % Determine the endpoints of the longest line segment
   len = norm(lines(k).point1 - lines(k).point2);
   if ( len > max_len)
      max_len = len;
      xy_long = xy;
   end
end
hold off


%To be able to plot the angles using the rose function, it is necessary to
transform all the degree values of theta to radians using deg2rad.
hold on
for k = 1:length(lines)
    lines(k).theta = - deg2rad(lines(k).theta);
end
hold off


%plotting the results (angles between 0-180)
figure;rose([lines.theta]);
```

MATLAB code used to automatically detect lines that were manually drawn on top of the images. The drawing had to follow certain rules for the program to work: it only detects straight lines, they must be of approximately similar length and width. A Windows laptop/tablet was used to draw on top of each image following these guidelines. Then, the Hough transformation creates a matrix where for each beginning and end of a line, it associates a vector to it and an orientation angle. The function ‘rose’ at the end of the code produces the rose diagrams representing the angle distribution of the vector lines.theta, it plots the different angles with respect to the reference being the vertical axis of the image set to 0 degrees. This process was repeated for all the images analysed without altering any parameter, to produce the angle distribution for each image.

Supplementary information 3
Estimation of the critical pore size to maintain plasma membrane integrity.

Our calculation is rooted in the elastic deformation energy of a membrane given by the Helfrich-Canham form[1],

$$F_H = \frac{\kappa}{2}\int H^2 dA, \qquad \text{eqn. 1}$$

where κ is the bending modulus of the membrane, H, the mean curvature, and dA the surface area element. We note that the spontaneous curvature of the membrane is assumed to be zero. Since the Gaussian curvature integrates to a constant term via the Gauss-Bonnet theorem for closed membranes the elastic deformation term corresponding to Gaussian curvature is neglected in Eq. 1. We visualize the cell membrane as a closed continuous sheet without any holes.

The mean curvature, $H = \frac{1}{2}(c_1 + c_2)$ where c_1 and c_2 are the principal curvatures of the membrane. For a hemispherical bulge of radius R the Helfich-Canham free energy is given by

$$F_H = \pi\kappa. \qquad \text{eqn. 2}$$

The work done to cause the membrane to deform into a hemispherical bulge will be given by $P\Delta V$ where P here is the turgor pressure of the cell and ΔV is the volume associated with the bulge. This gives a critical pressure to form the hemispherical bulge of

$$P_{crit} = \frac{3\kappa}{2R^3}. \qquad \text{eqn. 3}$$

At pressures above this we expect the membrane would start to escape out through the hole, ultimately leading to cell death. i.e. P_{crit} here is the maximum turgor pressure that the cell can sustain.

We calculate the expected maximum pore size for the expected cellular turgor pressure of approximately 20 bar. R is taken to be half the measured pore diameter. Values for the bending modulus of Gram positive cell membranes are hard to find in the literature. We estimate that κ is in the range 20-70 $k_B T$, the lower bound being the bending modulus for a lipid membrane, in line with Daly *et al*[2] and the upper bound being the bending modulus for a eukaryotic cell membrane lacking anchoring to the actin cortex[3]. For a turgor pressure of 20 bar, this gives us an estimated critical pore diameter of 8-12 nm, implying that the pores seen are just small enough that membrane rigidity is able to maintain cell turgor.

References

1. Helfrich, W. “Elastic Properties of Lipid BilayersTheory and Possible Experiments”, Z. Naturforsch. C 28 693-703 (1973)
2. Daly, E.K., Huang, K.C., Wingreen, N.S., Muckhopadhyay, R. “Mechanics of membrane bulging during cell-wall disruption in Gram-negative bacteria” Phys. Rev. E **83**, 041922 (2011)
3. Simson, R., Wallraff, E., Faix, J., Niewöhner, J., Gerisch, G., Sackmann, E. “Membrane bending modulus and adhesion energy of wil-type and mutant cells of Dictyostelium lacking Talin or Cortexillins” Biophys. J. **74**, 514-522 (1998)