Protocol for topology-preserving smoothing of BOLD fMRI retinotopic maps of the human visual cortex

The hierarchical organization of the visual system preserves topology, which is often lost in the "raw" human retinotopic maps derived from BOLD fMRI recordings. Here, we present the protocol for generating topology-preserving and smooth retinotopic maps from human retinotopy fMRI data. We describe data pre-processing, 3D surface flattening, and selection of the region of interest (ROI), followed by smoothing of the retinotopic maps within the ROI. This approach can be applied to visual cortical areas V1, V2, and V3 simultaneously.

Publisher’s note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.
Protocol

Protocol for topology-preserving smoothing of BOLD fMRI retinotopic maps of the human visual cortex

Yanshuai Tu,1,5 Xin Li,1,5 Zhong-Lin Lu,2,3,4,7,* and Yalin Wang1,6,*

1School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ, USA
2Division of Arts and Sciences, New York University Shanghai, Shanghai, China
3Center for Neural Science and Department of Psychology, New York University, New York, NY, USA
4NYU-ECNU Institute of Brain and Cognitive Science, NYU Shanghai, Shanghai, China
5These authors contributed equally
6Technical contact
7Lead contact
*Correspondence: zhonglin@nyu.edu (Z.-L.L.), ylwang@asu.edu (Y.W.)
https://doi.org/10.1016/j.xpro.2022.101614

SUMMARY

The hierarchical organization of the visual system preserves topology, which is often lost in the "raw" human retinotopic maps derived from BOLD fMRI recordings. Here, we present the protocol for generating topology-preserving and smooth retinotopic maps from human retinotopy fMRI data. We describe data pre-processing, 3D surface flattening, and selection of the region of interest (ROI), followed by smoothing of the retinotopic maps within the ROI. This approach can be applied to visual cortical areas V1, V2, and V3 simultaneously. For complete details on the use and execution of this protocol, please refer to Tu et al. (2021).

BEFORE YOU BEGIN

1. Data preparation. The protocol requires three inputs: cortical surface, retinotopic coordinates of the vertices on the surface, and initial visual area labels.
   a. The data from HCP (Human Connectome Project) have been preprocessed by the HCP group. Namely, the retinotopic coordinates, including the visual perception centers and sizes, have been decoded and projected onto the cortical surface. Therefore, downloading the following files will be sufficient:
      i. Files in the structural surface folder in the MNINonLinear fsaverage space that contain the cortical surface and initial visual area labels inferred from the registration step in HCP data preprocessing.
      ii. The retinotopic parameters from pRF decoding, which are stored in MAT-file (MATLAB DATA format) prfresults.mat.
   b. For data from other sources, our pipeline can also work. We recommend that the user pre-process the data to generate the following inputs for our pipeline:
      i. Cortical surface data in gii format.
      ii. Retinotopic coordinates of the vertices on the cortical surface.
      iii. Initial visual area labels, typically from FreeSurfer cortical surface registration.
2. Obtaining external packages.
   a. freesurferReader: This package provides utilities to read surface data generated by FreeSurfer.
   b. gifti: This package provides utilities to read data in gii format. This is the standard format used in the HCP data.
c. toolbox_fast_marching: This package provides utilities to perform geodesic distance computation.
d. pRF-decoder: This package is from http://kendrickkay.net/analyzePRF/. It’s used in pRF decoding and evaluation.
e. Geometry Processing Package: This package provides tools for reading, writing, and processing mesh data, including boundary specification and harmonic map computation used in the conformal mapping of the cortical surface.

**KEY RESOURCES TABLE**

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|----------------------|--------|------------|
| Software and algorithms |        |            |
| MATLAB (R2021a) | MathWorks website | mathworks.com |
| TPSR | Tu et al. (2021) | https://github.com/Retinotopy-mapping-Research/TPSR |
|  |  | https://zenodo.org/badge/latestdoi/473932553 |
| gifti | GitHub | https://github.com/gllmfndn/gifti.git |
| geometry-processing-package | MathWorks File Exchange | https://www.mathworks.com/matlabcentral/fileexchange/46540-geometry-processing-package |
| FreeSurfer and freesurferReader* | FreeSurfer | https://surfer.nmr.mgh.harvard.edu |
| toolbox_fast_marching | MathWorks File Exchange | https://www.mathworks.com/matlabcentral/fileexchange/6110-toolbox-fast-marching |
| analyzePRF** | Kendrick Kay | http://kendrickkay.net/analyzePRF/ |
| Deposited data | Human Connectome Project | https://www.humanconnectome.org/study/hcp-young-adult/article/first-release-of-7t-mr-image-data |
| Computer | NA | NA |

*The FreeSurfer package is used to process the MRI data, including image segmentation, cortical surface reconstruction and cortical surface registration. We recommend the user use it for data preprocessing if they use data from sources other than the HCP dataset.

**Our work takes pRF decoding results as the input. The analyzePRF package is used to generate the pRF decoding results and project them onto the cortical surface in the validation process.

**STEP-BY-STEP METHOD DETAILS**

**Organize the project folders and add the search paths**

© Timing: 2 min

This section mainly describes the recommended location of our package and outputs in user projects.

Note: We recommend that the user organize the project into several folders. The “code” folder contains our Topology-Preserving Smoothing of Retinotopic Maps (TPSR) package and third-party libraries, the “data” folder contains the necessary data for processing, and the “Figures” and “result” folders are optional and can be generated by the smoothing package automatically.

```
| - code/ // all the codes for the project.
| - Libraries/ // third party libraries.
| - TPSR/ // the core smoothing package.
| - data/ // data for validating our method.
```
1. Add “TPSR” to the MATLAB search path:

```matlab
> addpath(genpath('TPSR'));
```

2. Add the third-party packages:

```matlab
> addpath(genpath('libraries'));
```

3. Alternatively, the user can add search paths using the MATLAB user interface (Figure 1): Select the “libraries” folder, right click, and “Add to Path” with “Selected Folders and SubFolders”.

**Load and preprocess data**

© Timing: 5 min

This section introduces data preprocessing, including directly loading data from the OSF website or processing data from the HCP, flattening the cortical patch to a 2D disk, finding the region of interest on the disk, and establishing the visual coordinates of the boundary of the ROI.

4. Data preparation.
   a. Load data from our OSF website.

   **Note:** For users who wish to reproduce the results in Tu et al. (2021), we recommend that they directly download the “data/mesh_data” from our osf website (OSF data: https://doi.org/10.17605/OSF.IO/DBGKF) into “data/mesh”. The mesh data contain visual cortical surfaces cut from both hemispheres of the brain (Tu et al., 2021). Specifically, each cortical surface mesh has one boundary and contains the retinotopic coordinates and perceptual field sizes from the pRF solution and initial visual area labels.

   i. Load the mesh data with the “read_mfile” function in the < geometry-processing-package > library (In this example, the mesh file has 6K vertices, the computer has 8G RAM, and the loading takes less than 1 s):

   ```matlab
   > [Fm, Vm, Em]=read_mfile('..data/mesh_data/102311lh.m')
   ```

   **Note:** “Fm” is the face list, “Vm” is the vertex list. Extra information is stored in the “Em” struct, which contains pRF decoding result and visual area label for each vertex. [Note that the mesh data from our OSF site also contain pial surface coordinates, sphere registration coordinates, vertex id before cutting, etc.]

   ii. Some background on discrete surface: the fundamental data structure in surface processing is the triangular mesh, which consists of a list of vertices, a list of faces, and maybe (very often) some extra features associated with the vertices or faces.
b. Process data from the HCP.
For users who wish to start from a hemisphere of the brain in the HCP dataset, which is a genus zero sphere-like surface (therefore no boundaries), they need to cut a geodesic disk with the following procedure:

i. Find a center point roughly corresponding to the fovea of the primary visual cortex V1 (In our experiments, cortical surfaces are registered by FreeSurfer in the released HCP data. We are able to select a consistent point across subjects. There is probably some flexibility in center point selection, but more work is necessary to verify it.).

ii. Compute the geodesic distance between each point on the hemisphere and the center with the fast marching algorithm, which is similar to the Dijkstra algorithm that computes geodesic distance on graphs.

iii. Keep the patch made of points within a threshold geodesic distance (to include all points within the region of interest, e.g., radius=100). The final output, (Fcut, Vcut, id_cut), consists of the mesh of the geodesic disk and the index (before the deletion) of the kept vertices.

```matlab
start_points = 23452; % this point roughly corresponds to the fovea of V1 in the left hemisphere in the hcp 53k mesh
radius=100; % set the radius to keep
[D,S,Q]= perform_fast_marching_mesh(double(Vm), double(Fm),start_points);
ind2del = find(D > radius); % find points that are outside of the radius
[Fcut, Vcut, id_cut] = gf_remove_mesh_vertices(Fm, Vm, ind2del); % cut
```

5. Flatten the patch to a 2D disk.

Note: To simplify the smoothing process, we first convert the retinotopic map from the 3D cortical surface to a 2D disk. This involves flattening the 3D cortical surface onto a 2D planar disk in one-to-one mapping. We have written three functions to support the flattening process. The user can choose any one of them based on their own needs.

a. The function “disk_harmonic_map” flattens the patch with sufficient smoothness:

```matlab
> uv = disk_harmonic_map(Fcut, Vcut)
```

b. The function “disk_conformal_map” flattens the patch conformally (angle preserving):

```matlab
> uv = disk_conformal_map(Fcut, Vcut)
```

c. The function “disk_area_mapping” flattens the patch while preserving each local area:

```matlab
> uv = disk_area_mapping (Fcut, Vcut)
```

Note: We provide three different functions to parametrize the 3D visual cortical patch to the 2D domain to support different types of analysis of human retinotopic mapping. If the goal is to analyze angle distortions in retinotopic mapping, we recommend “disk_conformal_map” because the function does not introduce angle distortion in the flattening process. One may adopt “disk_area_mapping” function if the goal is to study metric properties in retinotopic mapping (e.g., cortical magnification factor). As a convenient and robust disk
parameterization method, the function, "disk_harmonic_map" may be used to perform preliminary qualitative analysis.

6. Draw the region of interest (ROI).

△ CRITICAL: The geodesic disk from steps 4a or 4b may contain too many vertices. In the example case, there are about 310 vertices. In this step, a region of interest (ROI) for smoothing is further cut from the disk with the following utilities (See Methods video S1).

a. If you start with our OSF data, please use the boundary provided in "data/V1.mat". Because the HCP 53k mesh data are registered and resampled with the same resolution, this boundary file can be applied to all the subjects in the HCP dataset.

b. If you start with the HCP data without registration or other data, the boundary of the ROI must be defined. We provide an interactive function, boundary_editor.m, for you to manually define the ROI.

Note: You can call this function directly in the MATLAB command line when the current path is the ‘code’ folder.

```matlab
% You can pass the file name as the parameter in this function. In our sample code, we use 102311lh.m
> boundary_editor('102311lh.m');
```

c. The program draws a visual polar angle mesh on the parametrized disk. Please left-click the mouse to define the mouse position as the starting vertex of the ROI polygon. Then right click the mouse on the disk to define other vertices of the ROI polygon.

d. After defining the ROI, a notice will appear in your command line. You need to input a file name to save the results.

```matlab
% Input the custom file name
> filename: result
```

Note: Then, the ‘result.mat’ file will be generated in your current path.

7. Prepare visual coordinates of the boundary.

a. If you start from our OSF data, the visual coordinates of the boundary are provided in “data/V1.mat”.

b. If you start from the HCP data without registration or other data, visual coordinates of the boundary are generated after you finish running “boundary_editor.m”.

Note: The visual coordinates of the boundary must be well-defined, with the winding number equal to 1. Otherwise, topological smoothing in step 9 will fail. We refer users to the troubleshooting section for further information. The visual coordinates of the boundary are stored in the variable “bd_vis0” after running the script “boundary_editor.m.”

8. Prepare the data for topological smoothing.

a. Prepare the face list of the ROI:

```matlab
[Froi, uv_roi, vfather] = gf_remove_mesh_vertices(Fcut, uv, id2delete);
```
b. Prepare the visual coordinates of the ROI:

```
prf = Em.Vertex_prf(vfather,:);
vis = prf(:, [2 1]) % The data are sorted in the order of eccentricity and polar angle
R2 = prf(:, 5)
```

c. Prepare the boundary id of the ROI:

```
bd_id = compute_bd(Froi);
```

d. Convert the parametric coordinates into a polar angle coordinate system by calling “uv_pol = cart2pol(uv_roi(:, 1), uv_roi(:, 2))”.

### Smooth the retinotopic maps

**Timing:** 1 min

This section describes the parameters used in the smoothing function and details of the smoothing process.

9. Use the function “topological_smoothing” to smooth the retinotopic map within the ROI:

```
> visxy_s = topological_smoothing(Froi, uv_pol, vis, R2, ...
    anchor, anchorpos, changetol, ...
    smooth_lambda0, smooth_avg_k, meanddth);
```

a. The parameters specified in the input to the function are:
   i. Froi: the face list of the ROI;
   ii. uv_pol: parametric coordinates of the ROI;
   iii. vis: retinotopic visual coordinates (eccentricity and polar angle) before smoothing;
   iv. R2: goodness of pRF decoding, ranging from 0 to 100;
   v. anchor: boundary index of the ROI (calculated in step 8);
vi. anchorpos: boundary visual coordinates of the ROI; 
vii. changetol: maximum change of boundary visual coordinates in smoothing; it is an empirical value. The user can customize it for different input data. 
viii. smooth_lambda0: smoothness factor, with higher values leading to more smoothing; 
ix. smooth_avg_k: number of neighbors of the boundary used to do average smoothing of the boundary if boundary smoothing is used during topological smoothing; 
x. meanddth: mean visual coordinates change allowed during smoothing. 

b. There are three sub-modules in this function:

i. Obtaining the boundary of the retinotopic map by extrapolating it from the interior of the ROI. We provide the function “smooth_bd_vis_byfit” to fit the interior of the ROI and predict new visual coordinates of the boundary.

```matlab
if changetol > 0
    for ti = 1:3
        bd_vis_new = smooth_bd_vis_byfit(uv_pol, vis, bd_id, R2);
        for i=1:size(bd_vis_new,1)
            bd_vid_change = bd_vis_new(i,:) - bd_vis0(i,:);
            if norm(bd_vid_change) > changetol
                bd_vis_new(i,:) = bd_vis0(i,:) + bd_vid_change/norm(bd_vid_change)*changetol; % update to the maximum tolerance
            end
        end
        bd_vis = bd_vis_new;
    end
else
    bd_vis = bd_vis0;
end

flip = get_flips(face, uv_pol, hat_vis);
if flip > 0
    % smooth and chop if not topological
    [hat_vis, flip] = make_diffeomorphic(face, uv_pol, hat_vis, bd_id, bd_vis);
end
```

Note: The new boundary for the first subject in the OSF data is shown in Figure 2 (the last segmentation of the boundary is not drawn to show the endpoints).

ii. Fixing the topology: 
Check the topological condition. It is well known from neurophysiology that retinotopic mapping is topological (i.e., the topology of neighborhood connectivity is preserved) within each visual area. In this study, we adopt the Beltrami coefficient, a metric of quasiconformal mapping, to define the topological condition, develop a mathematical model to quantify topological smoothing as a constrained optimization problem, and elaborate an efficient numerical method to solve
the problem. The topological condition is met when Beltrami coefficient \( \mu < 1 \) for all triangles. Re-check the topological condition and update the threshold if necessary. The smoothing process continues until the Beltrami coefficient \( \mu < 1 \) for all triangles.

```matlab
if size(hat_vis,1) ~= size(vis,1)
    hat_vis = vis;
end
end

> if(flip > 0)
    [~,flipid] = get_flips(face, uv_pol, hat_vis);
    % fail to fix topology, the only possible reason is that the boundary
    % value is not good, so we smooth on boundary value
    % smooth on boundary locally
    flipid_v = [face(flipid,1);
                face(flipid,2); face(flipid,3)];
    % feed flip vertices, and smooth on boundary near flips
    bd_vis_new = smooth_bd_vis_bybd(bd_id, bd_vis, smooth_avg_k, flipid_v);
    for i=1:size(bd_vis,1)
        bd_vid_change = bd_vis_new(i,:) - bd_vis(i,:);
        if norm(bd_vid_change) > changetol
            bd_vis_new(i,:) = bd_vis(i,:) + bd_vid_change/norm(bd_vid_change)*changetol;
            % update to the maximum tolerance
        end
    end
    bd_vis = bd_vis_new;
    % increase the strength of smoothing
    if changetol >= 0
        smooth_avg_k = smooth_avg_k +1;
    end
end
```

iii. Laplacian smoothing.

```matlab
A = (diag(R2) + smooth_lambda*laplace_beltrami(face,uv_pol));

b = (diag(R2)*hat_vis);

in = setdiff(1:size(uv_pol,1), bd_id);
hat_vis(in,:) = A(:,in)

hat_vis(bd_id,:) = bd_vis;
```
Note: An example is shown in Figure 3.

Note: The smoothing processing will terminate after the topological condition is satisfied.

Extension to V2 and V3

Timing: 1 min

Our algorithm is designed to work within one visual area. However, with some adjustments, the algorithm can be applied to V1, V2, and V3 simultaneously. More specifically, since the fovea of V1, V2 and V3 are very close, we introduce “extended polar angle”, which is continuous and can be obtained from a set of linear transformations of the polar angles in different visual areas, to remove the phase-jumping and changes of visual field signs for the left and right hemispheres in multiple visual areas. The specific transformations are explained in (Tu et al., 2021). We only list them here in Table 1.

In this way, the algorithm can be applied to V1, V2, and V3 simultaneously. We provide the following utilities to compute the extended polar angles.
10. Use the following utility to get the visual coordinates of the ROI:

```matlab
visxy_corrected = correct_vis(Em, 'lh'); % 'lh' stands for the left hemisphere
visxy_corrected = visxy_corrected(vfather,:);
```

11. Apply the same smoothing in step 9:

```matlab
visxy_s = topological_smoothing(Fcut, uv_pol, vis, R2,...
    anchor, anchorpos, changetol, ...)
    smooth_lambda0, smooth_avg_k, meanddth);
```

12. After smoothing, the function “restore_vis” can be used to transform the extended polar angles back to the original visual polar angles.

```matlab
vis_c = restore_vis(visxy_s);
```

**Note:** We provide an example, “GenFigure10_Smooth_V1V2V3.m”, for smoothing the retinotopic maps in V1, V2, and V3.

**EXPECTED OUTCOMES**

**Results on V1**
Smoothed visual coordinates (eccentricity and polar angle) of a V1 retinotopic map on the 2D surface (Figure 4), on the flattened cortical surface (Figure 5), and on the 3D cortical surface (Figure 6).

**Results on V1, V2, and V3**
Smoothed visual coordinates of a V1-V2-V3 complex retinotopic map (Figure 7): (a) the raw retinotopic map in the eccentricity-extended-polar coordinate space (with 232 overlapping triangles), (b) the smoothing results (no overlapping triangles), (c) the inflated mesh in medial view, (d) an enlarged picture of the raw retinotopic map on the inflated mesh, (e) the smoothing results on the inflated surface. The green and blue curves in (e) are levels sets of eccentricity and extended polar angle, respectively.

Because it contains visual area information, after smoothing the retinotopic maps of V1, V2, and V3, the extended polar angle can be used to delineate the interior boundaries of the
visual areas (Figure 8). In Figure 8, the colors of the surface represent manually labeled visual areas from the average data. The purple curves indicate boundaries between visual areas of V3d/V2d/V1d/V1v/V2v/V3v (from up to down), respectively. The green curves are eccentricity contours. Figure 9A: boundaries inferred from the raw data. Figure 9B: boundaries after topological smoothing.

QUANTIFICATION AND STATISTICAL ANALYSIS

We provide the function “evaluate_metric” to evaluate the smoothing results, including the average change of visual coordinates (“means”), the standard deviation of the change (“stdse”), the average change of eccentricity (“ecch”), and the average change of coordinates (“means”)

Figure 4. The smoothed V1 retinotopic map on the 2D surface

Figure 5. The smoothed V1 retinotopic map on the flattened cortical surface
angle distortion of the smoothed method ("mean_ang"), and the number of flipped triangles ("flip") to quantify the retinotopic map after smoothing. The number of flipped triangles monitors the topological condition of retinotopic maps.

```matlab
> [meanse, stdse, mean_ang, std_ang, flip] = evaluate_metric(visxy_s, vis, Froi, uv_pol);
```

This function returns the mean and standard deviation of value distortion, the mean and standard deviation of angle distortion, and the number of flipping triangles. "vn" is the smoothed/processed visual coordinates; "v" is the raw visual coordinates; "face" is the region-of-interest faces list, and "uv" is the coordinate in 2d of the region of interest.

Figure 6. The smoothed V1 retinotopic map on the cortical surface

Figure 7. The retinotopic map of the V1-V2-V3 complex of the left hemisphere of the first subject

Raw retinotopic maps displayed on the (A) 2D and (D) flatten cortical surfaces. Retinotopic maps after topological smoothing displayed on the (B) 2D surface, (C) 3D cortical surface, and (E) flattened cortical surface.
LIMITATIONS

Although we focus on the V1, V2, and V3 complex in this study, our method can in principle be extended to higher visual areas. However, there are some additional challenges. As we move to higher visual areas, the signal-to-noise ratio of the retinotopic map is further reduced. It is extremely challenging for smoothing methods to balance the topological condition and goodness of fit.

TROUBLESHOOTING

Problem 1

The boundary is critical for topological smoothing. The major task, fixing the topology, may fail for two reasons (Figure 9; step 9):

- The boundary of the region of interest is not a simple loop in the visual coordinate space, that is, the boundary of the region has a winding number greater than 1.
- When the resolution of mesh is low, the topological smoothing procedure may converge slowly because we conduct discrete operations on the mesh.

Potential solution

If the winding number is greater than 1, the boundary violates the topological condition. Caused by noise, this is not a real property of the retinotopic map. The solution is to smooth the boundary locally to make it a closed loop.

Figure 8. Visual area boundary delineation of the V1-V3
(A) Boundaries inferred from the raw retinotopic maps. (B) Boundaries after topological smoothing.

Figure 9. Winding number and closed loops
(A) A simple closed loop with winding number equal to 1. (B) and (C) Closed loops with winding number greater than 1.
If the resolution of the mesh is low, we can further reduce the magnitude of the Beltrami coefficient. For instance, if the Beltrami coefficient $m > 1$ for a triangle, the proposed method may shrink it by $m' = \beta m/|m|$ with a $\beta$ close but less than 1. To make the algorithm converge faster, we can set $\beta$ as a smaller value.

**RESOURCE AVAILABILITY**

**Lead contact**
Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Zhong-Lin Lu (zhonglin@nyu.edu).

**Materials availability**
This study will not generate new materials.

**Data and code availability**
Code and data are available through the GitHub repository: https://doi.org/10.5281/zenodo.6774984.

**SUPPLEMENTAL INFORMATION**
Supplemental information can be found online at https://doi.org/10.1016/j.xpro.2022.101614.

**ACKNOWLEDGMENTS**
Y.T., Z.L., and Y.W. were partially supported by the Directorate for Mathematical and Physical Sciences of the National Science Foundation, grant no. 1413417 to D.T., Y.T., and Y.W., and grant no. 1412722 to Z.L. Y.T., X.L., Z.L., and Y.W. were partially supported by National Eye Institute, grant no. R01EY032125. Y.T. and Y.W. were partially supported by the National Institute on Aging, grant no. R21AG065942; the National Institute of Biomedical Imaging and Bioengineering, grant no. R01EB025032; National Institute of Dental & Craniofacial Research, grant no. R01DE030286;
and Arizona Alzheimer’s Consortium. The funders had no role in the study design, data collection, and analysis, decision to publish, or preparation of the manuscript.

**AUTHOR CONTRIBUTIONS**
Y.T.: methodology, conceptualization, software, original draft preparation. X.L.: software organization and management, manuscript draft. Z.L.L.: methodology, supervision, review and editing, funding. Y.W.: methodology, supervision, review and editing, project administration, funding.

**DECLARATION OF INTERESTS**
Y.T., Z.L., and Y.W. have a joint patent application: Tu, Y., Y. Wang, and Z.-L. Lu, Methods and Systems for Precise Quantification of Human Sensory Cortical Areas, U.S. Patent Application No. 63/004. 2020.

**REFERENCES**
Tu, Y., Ta, D., Lu, Z.L., and Wang, Y. (2021). Topology-preserving smoothing of retinotopic maps. PLoS Comput. Biol. 17, e1009216. https://doi.org/10.1371/journal.pcbi.1009216.