

## A unified view on beamformers for M/EEG source reconstruction

Britta U. Westner<sup>a,b,\*</sup>, Sarang S. Dalal<sup>b</sup>, Alexandre Gramfort<sup>c</sup>, Vladimir Litvak<sup>d</sup>,  
John C. Mosher<sup>e</sup>, Robert Oostenveld<sup>a,f</sup>, Jan-Mathijs Schoffelen<sup>a</sup>

<sup>a</sup> Radboud University, Donders Institute for Brain, Cognition and Behaviour, Nijmegen, The Netherlands

<sup>b</sup> Center of Functionally Integrative Neuroscience, Department of Clinical Medicine, Aarhus University, Aarhus, Denmark

<sup>c</sup> Université Paris-Saclay, Inria, CEA, Palaiseau, France

<sup>d</sup> Wellcome Centre for Human Neuroimaging, UCL Queen Square Institute of Neurology, London, UK

<sup>e</sup> Texas Institute for Restorative Neurotechnologies, McGovern Medical School, University of Texas Health Science Center at Houston, TX USA

<sup>f</sup> NatMEG, Karolinska Institutet, Stockholm, Sweden

### ARTICLE INFO

#### Keywords:

MEG  
EEG  
Data analysis  
Source reconstruction  
Source imaging  
Source localization  
Beamforming

### ABSTRACT

Beamforming is a popular method for functional source reconstruction using magnetoencephalography (MEG) and electroencephalography (EEG) data. Beamformers, which were first proposed for MEG more than two decades ago, have since been applied in hundreds of studies, demonstrating that they are a versatile and robust tool for neuroscience. However, certain characteristics of beamformers remain somewhat elusive and there currently does not exist a unified documentation of the mathematical underpinnings and computational subtleties of beamformers as implemented in the most widely used academic open source software packages for MEG analysis (Brainstorm, FieldTrip, MNE, and SPM). Here, we provide such documentation that aims at providing the mathematical background of beamforming and unifying the terminology. Beamformer implementations are compared across toolboxes and pitfalls of beamforming analyses are discussed. Specifically, we provide details on handling rank deficient covariance matrices, prewhitening, the rank reduction of forward fields, and on the combination of heterogeneous sensor types, such as magnetometers and gradiometers. The overall aim of this paper is to contribute to contemporary efforts towards higher levels of computational transparency in functional neuroimaging.

### 1. Introduction

Initially created for sonar and radar applications, the beamforming technique has been introduced into neuroscience as a method for interpreting the neural basis of MEG and EEG data (van Dronkelen et al., 1996; Van Veen and Buckley, 1988; Van Veen et al., 1997). Since its first implementation in neuroscientific open source toolboxes (namely, Nutmeg [Dalal et al., 2004; Dalal et al., 2011] and FieldTrip [Oostenveld et al., 2011]), beamforming has become a widely applied source reconstruction technique in the field and is now implemented in various M/EEG signal processing software packages. Features such as the ability to resolve deeper sources (Backus et al., 2016; Quraan et al., 2011; Wilson et al., 2010) or to suppress external noise (Litvak et al., 2010; Sekihara et al., 2004) make them widely used for the source analysis of M/EEG data. However, due to the rather independent development of open source beamformer implementations, there are considerable discrepancies in both the nomenclature and implementational details across toolboxes. While detailed resources on the mathematical foundations of beamformers for neuroscience exist (e.g., Hillebrand and Barnes, 2005; Sekihara and Nagarajan, 2008; 2015), it is not al-

ways straightforward to link these to the beamformer implementations available in M/EEG toolboxes that are frequently used in practice. In addition, in practical applications, recording device dependent data properties require specific processing of the data to optimize the analysis.

This paper thus gives a mathematical and conceptual overview of the most frequently used beamformer types and aims at unifying the terminology of beamforming in neuroscience. Furthermore, we detail the commonalities and differences regarding the user interfaces and computation of different beamforming algorithms with respect to four major open source toolboxes: Brainstorm (Mosher et al., 2005; Tadel et al., 2011), FieldTrip (Oostenveld et al., 2011), MNE-Python (Gramfort et al., 2013, 2014), and SPM (Litvak et al., 2011). Lastly, we describe some practical considerations with respect to optimal data handling in beamforming analyses, and describe best practices for source reconstructing M/EEG data with beamformers.

### 2. Overview of beamformer types

As with other source reconstruction techniques such as minimum norm estimation (Hämäläinen and Ilmoniemi, 1994), beamformers pro-

\* Corresponding author at: Radboud University, Donders Institute for Brain, Cognition and Behaviour, Nijmegen, the Netherlands.

E-mail address: [britta.westner@donders.ru.nl](mailto:britta.westner@donders.ru.nl) (B.U. Westner).

vide a link between two spatial domains. On the one hand there is the “channel” space in which the measurements are done, on the other hand there is the “source” space where we want to interpret the data. To describe this transformation from channel to source space, the terms “source modelling”, “estimation”, “reconstruction”, “localization”, “M/EEG brain imaging”, and “unmixing” are often used interchangeably. These concepts all relate to each other, but each term puts a slightly different emphasis on the intended goal of the beamforming procedure. For instance, “localization” focuses more on the identification of the location of the underlying neuronal sources, and thus implies an interpretation of a spatial image, for instance with respect to local maxima. In contrast, “unmixing” puts more emphasis on the temporal characteristics of the sources, implying an interpretation in terms of the dynamics of the underlying processes.

Beamforming is typically done by scanning through a set of predefined putative source locations, computing separately for each location a set of weights, a so-called spatial filter, and applying these weights to the measured sensor data to obtain the beamformer output for each location. Thus, the spatial filter reflects the individual sensor’s contribution to each location’s source estimate (Baillet et al., 2001; Hillebrand and Barnes, 2005; Van Veen et al., 1997). Assuming a focal source modeled as an equivalent current dipole with a freely defined orientation (*i.e.*, a regional source), the beamformer output  $\hat{\mathbf{S}}$  is an estimate of the dipole moment at location  $r$  and time point  $t$ , and is a 3-dimensional vector that can thus be expressed as

$$\hat{\mathbf{S}}(r, t) = \mathbf{W}^\top(r) \mathbf{b}(t) \quad (1)$$

where the  $M \times 3$  matrix  $\mathbf{W}$  is the spatial filter, with  $M$  being the number of sensors and the three columns referring to the  $x$ ,  $y$ , and  $z$  components of the dipole moment.  $\mathbf{b}(t)$  is the  $M \times 1$  vector of measured channel data at time point  $t$ . Such a beamformer is called a vector beamformer. Alternatively, when assuming a fixed orientation dipole at location  $r$ , the spatial filter  $\mathbf{w}$  will be a  $M \times 1$  vector for each source location. Because each sensor signal is weighted with a single weight to generate a scalar value for the beamformer output at each time point, such a beamformer is called a scalar beamformer. While Eq. (1) describes the mapping from sensor to source space for a vector beamformer, the scalar equivalent can be expressed as:

$$\hat{s}(r, t) = \mathbf{w}^\top(r) \mathbf{b}(t) \quad (2)$$

The measured channel data will play a crucial role in the computation of the spatial filter  $\mathbf{W}$  or  $\mathbf{w}$  (see Section 2.1), in the form of the data covariance matrix  $\mathbf{R}$ :

$$\mathbf{R} = (\mathbf{B}\mathbf{B}^\top)/T \quad (3)$$

where  $\mathbf{B}$  refers to the  $M \times T$  time series of the sensor space data over  $T$  time samples of interest. The covariance matrix  $\mathbf{R}$  represents the pairwise co-variation of sensor signals.

Considering the beamformer output over time, it is custom to also consider its covariance, which is expressed as:

$$\text{cov}(\hat{\mathbf{S}}(r)) = (\hat{\mathbf{S}}(r)\hat{\mathbf{S}}(r)^\top)/T = \mathbf{W}^\top[(\mathbf{B}\mathbf{B}^\top)/T]\mathbf{W} = \mathbf{W}^\top\mathbf{R}\mathbf{W} \quad (4)$$

For a vector beamformer, this quantity reflects the covariance of the local dipole with unconstrained orientation, and can for example be used to compute the optimal source orientation, as we will see in Section 2.1.

Any beamformer output can be used to generate scalar spatial maps for subsequent analysis, for instance to generate experimental contrasts. For vector beamformer output time series, the three values per source localization need to be combined, *e.g.*, by taking the root mean square, or the sum of the diagonal elements (the trace) for the source covariance, or by retaining the first principal component.

Beamformer types differ in the way in which the spatial filter matrix  $\mathbf{W}$  (or vector  $\mathbf{w}$ ) is computed. In the following, we will address commonly used and implemented beamformer variants. We will first introduce the basic equations for the unit-gain beamformer introduced by Van Veen and colleagues (1988; 1997). In Section 2.2, we will look at

different spatial normalization strategies, which result in slightly different formulations for the spatial filter. This is followed by a brief discussion of the dynamic imaging of coherent sources (DICS) beamformer (Gross et al., 2001), which operates on data that is defined in the frequency domain.

### 2.1. Basic beamformer formulations

The original linearly-constrained minimum variance (LCMV) beamformer for M/EEG data (van Drongelen et al., 1996; Van Veen et al., 1997) computes the spatial filter weights for a vector beamformer as follows (*cf.* Sekihara and Nagarajan, 2008; 2015; Van Veen et al., 1997):

$$\mathbf{W}^\top(r) = [\mathbf{L}^\top(r)\mathbf{R}^{-1}\mathbf{L}(r)]^{-1}\mathbf{L}^\top(r)\mathbf{R}^{-1} \quad (5)$$

Here,  $\mathbf{R}$  denotes the  $M \times M$  covariance matrix of the sensor space data, and  $\mathbf{L}(r)$  denotes the  $M \times 3$  forward field matrix at location  $r$  for directions  $x$ ,  $y$ , and  $z$ . The forward field matrix represents the solution to the forward problem, describing the magnetic fields or electrical potentials outside the head, arising from dipolar sources in the different locations inside the head. More specifically, the first column of  $\mathbf{L}(r)$  contains the sensor activity that would be produced by a source at location  $r$  with unitary strength in the  $x$  orientation; similarly, the second and third columns are for  $y$  and  $z$  orientations, respectively. The forward field is determined by the location of the source, the sensor or electrode configuration, and the volume conductor model. The forward field is also commonly referred to as the gain matrix or lead field, although the latter in the strict sense only refers to the relationship among many sources and one electrode or sensor. It is evident from Eq. (5), that the spatial filter  $\mathbf{W}$  depends solely on the forward field matrix and the inverse of the data covariance matrix.

The above formula represents a vector beamformer; the scalar case is given by:

$$\mathbf{w}^\top(r) = [\mathbf{l}^\top(r)\mathbf{R}^{-1}\mathbf{l}(r)]^{-1}\mathbf{l}^\top(r)\mathbf{R}^{-1} \quad (6)$$

Now, the  $M \times 1$  vector  $\mathbf{l}(r) = \mathbf{L}(r)\boldsymbol{\eta}(r)$  describes the forward field for a specific orientation  $\boldsymbol{\eta}$  at location  $r$ . Consequently, the beamformer output per location is now a scalar, representing only the orientation specified by the forward field. The orientation can be determined by different approaches, *e.g.*, using a fixed orientation orthogonal to the local cortical surface (Dale and Sereno, 1993). A frequently used version of the scalar beamformer considers a data derived optimal orientation  $\boldsymbol{\eta}_{\text{opt}}$  which is the one that maximizes the output power of the source estimate. This can be done by an eigenvalue decomposition of the output source covariance  $\text{cov}(\mathbf{S}) = (\mathbf{L}^\top\mathbf{R}^{-1}\mathbf{L})^{-1}$ , which is obtained by inserting the definition of  $\mathbf{W}$  from Eq. (5) into Eq. (4). The orientation that maximizes the output is the eigenvector corresponding to the largest eigenvalue. Mathematically equivalently, the orientation  $\boldsymbol{\eta}_{\text{opt}}$  can be efficiently computed as:

$$\boldsymbol{\eta}_{\text{opt}}(r) = \boldsymbol{\vartheta}_{\min}\{\mathbf{L}^\top(r)\mathbf{R}^{-1}\mathbf{L}(r)\} \quad (7)$$

where  $\boldsymbol{\vartheta}_{\min}\{\cdot\}$  denotes the eigenvector corresponding to the smallest eigenvalue.

The obtained orientation can then be used in the computation of the scalar beamformer. Another, yet slightly different, heuristic to estimate an optimal orientation was implemented in the original version of the synthetic aperture magnetometry (SAM) beamformer (Robinson and Vrba, 1999; Vrba and Robinson, 2000). This beamformer originally determined the optimal orientation by evaluating the scalar spatial filter output power for a large number of dipole orientations, and then selected the orientation that corresponds to the largest power. Although different in its algorithmic approach, the SAM beamformer will yield similar results to the orientation selection described in Eq. (7).

The beamformers described in Eq. (5) (vector case) and Eq. (6) (scalar case) are both referred to as unit-gain beamformers. They impose a unit-gain constraint  $\mathbf{w}^\top(r)\mathbf{l}(r) = 1$ , or  $\mathbf{W}^\top(r)\mathbf{L}(r) = \mathbf{I}$  for

the vector case, with  $\mathbf{I}$  denoting the identity matrix. This can be verified by plugging the expressions for the spatial filters from Eqs. (5) and (6) into the equations for these constraints. A unit gain means that the signal from the source at  $r$  is passed without attenuation. At the same time, assuming that the underlying sources are temporally uncorrelated, contributions to the estimated signal at location  $r$  from sources at any location other than  $r$  are minimized. This latter feature reflects the minimum variance constraint that is referred to in the LCMV acronym mentioned above.

The formulas for all beamformer variants that will be introduced in this paper have a similar structure. The formula consists of two terms, one of which is a matrix inverse. The term  $\mathbf{L}^T(r)\mathbf{R}^{-1}$  (vector case) or  $\mathbf{I}^T(r)\mathbf{R}^{-1}$  (scalar case) is identical in all the variants we discuss here. Note, however, that the heuristic that may be used for the computation of the optimal orientation might be different across beamformer types, which has a non-trivial impact on the beamformer output (e.g., in comparing the unit-gain beamformer with the unit-noise-gain beamformer; see Section 2.2). Furthermore, the inverted matrix term that differs among the beamformer variants eventually determines the physical units in which the beamformer output is expressed. For instance, the unit-gain beamformer (Eq. (5) and (6)) results in output that has physical units of the dipole moment (nAm), while the unit-noise-gain beamformer discussed in 2.2 yields outputs often referred to as “z-scores”, scaled to the standard deviation of noise.

## 2.2. Spatial normalization strategies

The magnitude of forward field coefficients decreases with the distance of the dipole (its depth) to the sensors. As can be seen from Eqs. (5) and (6), the magnitude of the unit-gain beamformer weights, and hence the magnitude of the corresponding beamformer output, depends on the inverse of the forward field matrix at a given dipole location. This so-called *center of head bias* causes the reconstructed power of sources that are far away from the sensors to be a few orders of magnitude larger than the reconstructed power of more superficial sources. This places restrictions on a direct magnitude-based comparison of the beamformer output across space.

Therefore, in order to be able to interpret the spatial patterns of the beamformer reconstruction, it is necessary to apply a location-specific normalization. This can be achieved in various ways. On the one hand, it is possible to apply a normalization as a post-processing step in the analysis procedure, i.e., after the computation of the beamformer output. Examples of these post-hoc normalization strategies include the computation of a neural activity index (NAI; Van Veen et al., 1997), a pseudo-Z statistic (Robinson and Vrba, 1999; Vrba and Robinson, 2000) or the evaluation of an experimental contrast. Alternatively, it is possible to use a beamformer formulation in which a normalization is achieved at the spatial filter computation step. In the following, we describe two beamformer types which achieve this spatial filter normalization: the array-gain beamformer (using forward field normalization) and the unit-noise-gain beamformer (using weight normalization).

**Array-gain beamformers** The scalar array-gain beamformer implements the constraint  $\mathbf{w}^T(r)\mathbf{l}(r) = \|\mathbf{l}(r)\|$ , where  $\|\mathbf{l}(r)\|$  is defined as the forward field vector’s norm. This constraint is equivalent to  $\mathbf{w}^T(r)\mathbf{l}(r)/\|\mathbf{l}(r)\| = 1$ , which implies that if the forward field is normalized such that  $\mathbf{I}_n = \mathbf{l}(r)/\|\mathbf{l}(r)\|$ , we can use the formula for a standard unit-gain beamformer (cf. Eq. (6)) to derive the spatial filter:

$$\mathbf{w}^T(r) = [\mathbf{I}_n^T(r)\mathbf{R}^{-1}\mathbf{I}_n(r)]^{-1}\mathbf{I}_n^T(r)\mathbf{R}^{-1}, \quad (8)$$

with  $\mathbf{I}_n$  denoting the normalized forward field. Due to the normalization of the forward field vector, the spatial filter does not have physical units, and thus the beamformer output is expressed in the units of the measurement (e.g., T or  $\mu\text{V}$ ).

In the vector case, the array-gain constraint has been defined as (Sekihara and Nagarajan, 2008):

$$\mathbf{W}^T(r)\mathbf{L}(r) = \begin{bmatrix} \|\mathbf{l}_x(r)\| & 0 & 0 \\ 0 & \|\mathbf{l}_y(r)\| & 0 \\ 0 & 0 & \|\mathbf{l}_z(r)\| \end{bmatrix}. \quad (9)$$

This implies the forward field matrix to be normalised for each cardinal orientation separately:

$$\mathbf{L}_n(r) = \begin{bmatrix} \mathbf{l}_x(r) & \mathbf{l}_y(r) & \mathbf{l}_z(r) \\ \|\mathbf{l}_x(r)\| & \|\mathbf{l}_y(r)\| & \|\mathbf{l}_z(r)\| \end{bmatrix}. \quad (10)$$

The weights are again computed using the normalized forward field  $\mathbf{L}_n$ :

$$\mathbf{W}^T(r) = [\mathbf{L}_n^T(r)\mathbf{R}^{-1}\mathbf{L}_n(r)]^{-1}\mathbf{L}_n^T(r)\mathbf{R}^{-1}. \quad (11)$$

It has been noted, however, that with this formulation the beamformer output is not rotationally invariant (Lalancette, 2014), which means that the output depends on the (arbitrarily defined) coordinate system in which spatial coordinates are expressed. This might not be desirable and we propose to use an alternative definition of the array-gain constraint:

$$\mathbf{W}^T(r)\mathbf{L}(r) = \|\mathbf{L}(r)\|_F, \quad (12)$$

which equivalently means a normalization of the forward field matrix by its Frobenius norm. By using this scalar normalisation, the resulting beamformer is rotationally invariant, but at the same time ensures an array gain independent of source depth. This latter version of the vector array-gain beamformer is implemented in FieldTrip and MNE-Python.

**Unit-noise-gain beamformers** Another spatial normalization strategy is implemented in the unit-noise-gain beamformer, which is also referred to as the weight-normalized beamformer or the Borgiotti-Kaplan beamformer, as it was originally proposed by Borgiotti and Kaplan (1979). This variant uses two constraints for the spatial filter calculation,  $\mathbf{w}^T(r)\mathbf{l}(r) = \tau$  and  $\mathbf{w}^T(r)\mathbf{w}(r) = 1$ . It is the first constraint that is used to solve the minimization problem (cf. Borgiotti and Kaplan, 1979), with  $\tau$  being determined by the second constraint. The second constraint intends to achieve the noise gain to be unity, under the assumption that the sensor noise is spatially uncorrelated and equally distributed across channels.

Adopting the above-mentioned constraints, the spatial filter weights for the scalar unit-noise-gain beamformer are defined as (cf. Sekihara and Nagarajan, 2008; 2015):

$$\mathbf{w}^T(r) = [\mathbf{I}^T(r)\mathbf{R}^{-2}\mathbf{I}(r)]^{-\frac{1}{2}}\mathbf{I}^T(r)\mathbf{R}^{-1}. \quad (13)$$

Another, perhaps more intuitive, way to look at this beamformer type is to examine its relation to the unit-gain beamformer described in Eq. (6). With the unit-gain beamformer weights as described in Eq. (6) and here denoted as  $\mathbf{w}_{\text{ug}}$ , the unit-noise-gain beamformer weights  $\mathbf{w}_{\text{ung}}$  can be derived as

$$\mathbf{w}_{\text{ung}}^T = \frac{\mathbf{w}_{\text{ug}}}{\|\mathbf{w}_{\text{ug}}\|}, \quad (14)$$

which illustrates why this beamformer type is also referred to as a weight-normalized beamformer.

The optimal orientation for this spatial filter is computed as follows (cf. Sekihara and Nagarajan, 2008):

$$\mathbf{n}_{\text{opt}}(r) = \mathfrak{D}_{\min}\{\mathbf{L}^T(r)\mathbf{R}^{-2}\mathbf{L}(r), \mathbf{L}^T(r)\mathbf{R}^{-1}\mathbf{L}(r)\}, \quad (15)$$

with  $\mathfrak{D}_{\min}\{\cdot, \cdot\}$  representing the eigenvector that is corresponding to the minimum generalized eigenvalue of the two matrices.

Although a vector version of this beamformer type has been defined by Sekihara and Nagarajan (2008, Eq. 4.83 and 4.85), it has been noted that this definition is not rotationally invariant (Lalancette, 2014) and therefore we do not recommend its usage.

### 2.3. Frequency-domain beamformers

Neural activity often manifests itself as (modulations of) rhythmic components in the signal, as opposed to evoked responses; such modulations often correlate well with underlying multiunit activity or mediate connectivity among brain regions (e.g., Buzsaki, 2006; Singh, 2012). To facilitate analysis of frequency-resolved oscillatory activity in source space as well as coherence among sources, Gross et al. (2001) introduced a variant of the LCMV beamformer, the dynamic imaging of coherent sources (DICS) method.

For the vector version of this beamformer, the weights  $\mathbf{W}$  (or  $\mathbf{w}$ ) at a specific frequency of interest  $f$  and source location  $r$  are computed as follows:

$$\mathbf{W}^\top(r, f) = [\mathbf{L}^\top(r)\mathbf{Q}(f)^{-1}\mathbf{L}(r)]^{-1}\mathbf{L}^\top(r)\mathbf{Q}(f)^{-1}, \quad (16)$$

where the  $M \times M$  matrix  $\mathbf{Q}(f)$  describes the cross-spectral density (CSD) matrix for frequency  $f$  or a frequency band centered around  $f$  (Gross et al., 2001). Note how Eq. (16) can be obtained by simply replacing the covariance matrix  $\mathbf{R}$  with the CSD matrix  $\mathbf{Q}(f)$  in Eq. (5).

The CSD matrix describes the shared power and phase shift between two signals at a given frequency, here between the signals of the different sensors, and is a complex-valued matrix:

$$\mathbf{Q}(f) = \mathbf{X}(f)\mathbf{X}^H(f), \quad (17)$$

where  $\mathbf{X}(f)$  denotes the frequency domain sensor space signals at a given frequency  $f$  (or a frequency band centered around  $f$ ) and  $\mathbf{X}^H$  refers to the Hermitian transpose (the complex conjugate transpose) of matrix  $\mathbf{X}$ .

Application of the complex-valued CSD matrix in Eq. (16) yields a complex-valued spatial filter, which does not have a valid biophysical interpretation. The consequence of complex numbers in the spatial filter is that the contributions of individual channels are shifted in phase relative to one another and in relation to the assumed source, which violates the linear instantaneous mapping from sources to the channels (i.e., without phase shifts). For this reason it is customary to only consider the real-valued part of the CSD matrix in the computation of the spatial filter.

The DICS beamformer output provides the spectral power estimates for frequency  $f$  for every predefined source location. Moreover, the DICS beamformer allows for an efficient computation of the spectral coherence between pairs of dipolar sources, or between dipolar sources and an external reference channel (e.g., the electromyogram (EMG) as in Schoffelen et al., 2005). For example, the cross-spectral density between two source locations  $r_i$  and  $r_j$ ,  $\hat{\mathbf{C}}(r_i, r_j, f)$ , can be computed by using the spatial filters from two different locations, in combination with the sensor level cross-spectral density, as follows:

$$\hat{\mathbf{C}}(r_i, r_j, f) = \mathbf{W}^H(r_i, f)\mathbf{Q}(f)\mathbf{W}(r_j, f), \quad (18)$$

with  $\mathbf{W}(r_i, f)$  and  $\mathbf{W}(r_j, f)$  being the  $M \times 3$  weight matrices for frequency  $f$  at source location  $r_i$  and  $r_j$  and  $\mathbf{Q}(f)$  denoting the (complex-valued) CSD matrix of frequency  $f$ .  $\hat{\mathbf{C}}(r_i, r_j, f)$  is the cross-spectral density between the two source locations, which can subsequently be used to compute a spectral interaction measure between the two sources, e.g., the magnitude squared coherence, or the imaginary part of coherence.

Eq. (16) illustrates the close similarity between the LCMV and DICS weight calculation. While this section only presents the unit-gain vector case of the DICS beamformer, as per its original formulation by Gross et al. (2001), the principles and variations shown in Section 2.2 with respect to the LCMV beamformer can be applied to the DICS beamformer as well.

### 3. Practical considerations and best practices in beamformer source reconstruction

#### 3.1. Estimation and inversion of covariance matrices

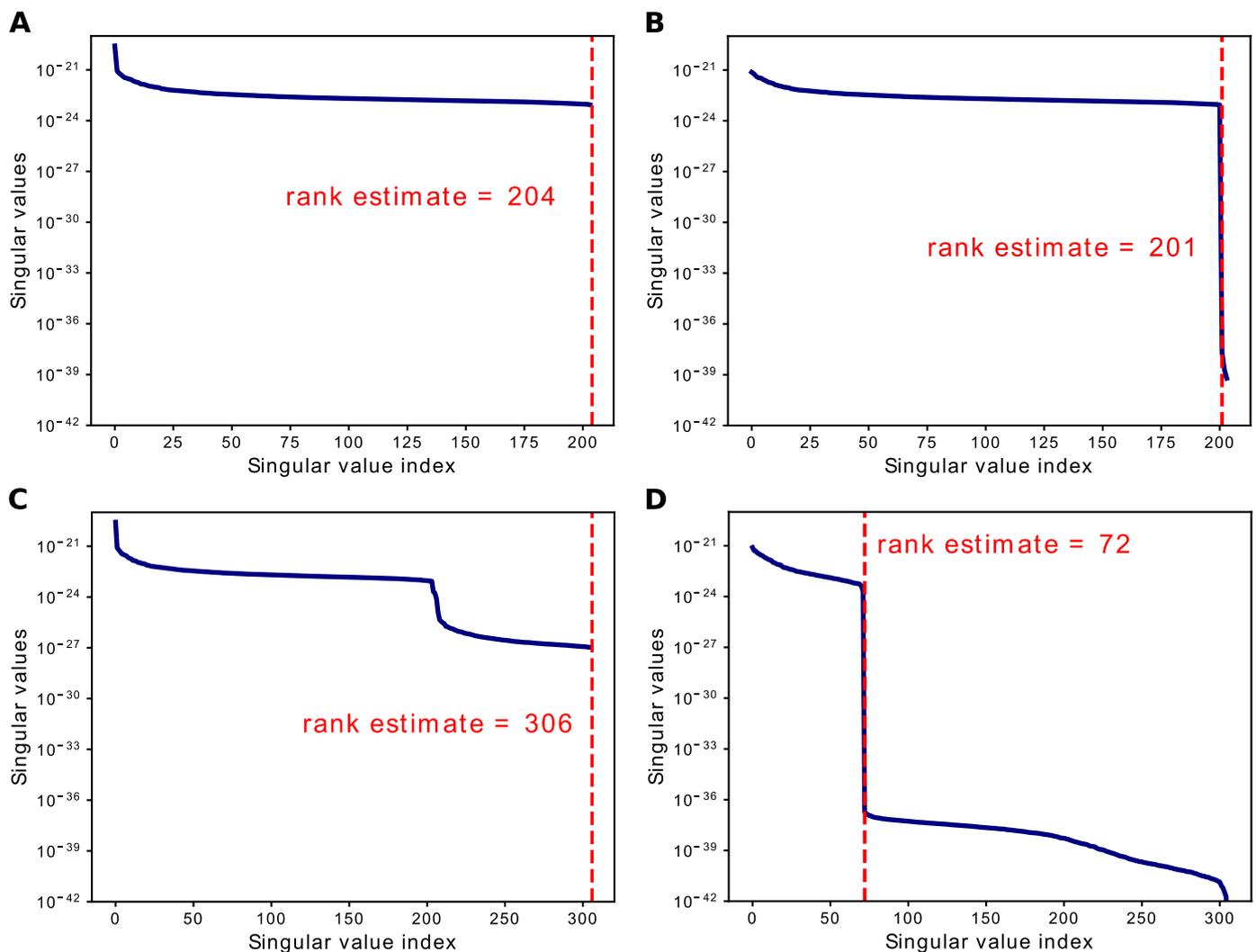
*Estimation and inspection of covariance matrices* As pointed out in Section 2.1, the spatial filter depends on the forward field and the mathematical inverse of the data covariance matrix (e.g., Eq. (5)). Consequently, a high quality of both ingredients is crucial for beamformer performance. The covariance is obtained from the signals and needs to be estimated from the limited data that is available. Given  $M$  sensors this means one needs to estimate  $M(M+1)/2$  unique entries in an  $M \times M$  matrix.

Computing an estimate of  $\mathbf{R}$  can be done by simply estimating the pairwise channel covariance on a given window of interest. This is the so-called maximum likelihood estimator. Due to hardware limitations, early implementations may have estimated the covariance from averaged data, but otherwise it is best practice to estimate the covariance from concatenated data. This increases the number of samples from which the numerous coefficients in  $\mathbf{R}$  can be computed, hence improving the quality of the estimation. It also decreases the chance of highly correlated sources by preserving trial-to-trial differences in signal and noise, consequently decreasing the chance of distorted beamformer output (see for example Van Veen et al., 1997).

To deal with the statistical challenge of estimating  $\mathbf{R}$  robustly from limited numbers of samples, several enhanced estimation methods exist, e.g., probabilistic principal component analysis (Tipping and Bishop, 1999), minimum covariance determinant estimation (Rousseeuw and Driessen, 1999) or the Ledoit-Wolf shrinkage model (Ledoit and Wolf, 2004a). Cross-validation can help in selecting the best model for covariance estimation (Engemann and Gramfort, 2015).

Depending on the experimental context, and details of the pre-processing applied to the channel data, the quality of the covariance matrix may however be limited. Two summary statistics that quantify the overall quality of the covariance matrix are the condition number and the effective rank. The rank is a matrix property that describes the maximum number of linearly independent columns that is needed to fully describe the matrix, and a matrix is rank-deficient if the number of independent dimensions is less than the number of rows/columns. In practice this means that a matrix is rank deficient if the rank is less than the number of channels. The condition number, defined as the ratio of the largest to smallest singular value of a matrix, quantifies the range of the magnitudes with which the independent linear dimensions are represented in the matrix. If this range is large (i.e., if it spans the magnitude range of the computer's numerical precision), small components, which often reflect noise in the estimation, will negatively impact the matrix inversion.

Ill-conditioned covariance matrices can arise when estimated over particularly small amounts of data. This adheres to too few and too short data segments (especially if the number of sensors is larger than the number of time samples), but the signal bandwidth plays a role as well or if the data has an overly narrow bandwidth (cf. Brookes et al., 2008; Dalal et al., 2008). While this can be mitigated through proper experimental design by maximizing the amount of collected data (Brookes et al., 2008; Gross et al., 2013), ill-conditioned covariance matrices can also occur if the data is preprocessed in a manner that makes the data linearly dependent across channels. This is for example the case for EEG data, as the signals at each electrode are recorded relative to another electrode. Since the reference electrode is present in all other channels, the data is rank-deficient by this one channel. Another example is the use of Independent Component Analysis (ICA) on the data for artifact suppression: after the removal of components, the data is rank-deficient by the number of rejected components. Signal Space Separation (SSS, e.g., MEGIN's MaxFilter) is another spatial cleaning algorithm that removes components from the data that originate from outside of the head



**Fig. 1.** The figure panels all show the singular value spectrum (dark blue trace) of the data covariance matrix of one data set after different data pre-processing procedures. The rank estimate (red dashed line) is obtained through singular value decomposition. The SVD spectra all show a cliff at their effective rank. **A** Singular value spectrum of a full-rank data covariance matrix with 204 sensors. **B** Singular value spectrum of the covariance matrix in **A** after removing 3 ICA components. **C** Singular value spectrum of a full-rank covariance matrix with two sensor types (102 magnetometers and 204 gradiometers), note the additional, smaller cliff due to the two sensor types. **D** The same data as in **C** but after applying Signal Space Separation. The data used for all figures is the “sample data set” from MNE-Python, which is a dataset with auditory and visual stimulation (Gramfort, Luessi, Larson, Engemann, Strohmeier, Brodbeck, Parkkonen, Hämäläinen, 2014). The figures shown here all rely on the subset of the data with stimulation in the right visual field. The code to reproduce the figures can be retrieved from [https://github.com/britta-wstnr/beamformer\\_examples](https://github.com/britta-wstnr/beamformer_examples). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(Taulu et al., 2003), which can result in the loss of several tens of linear components (see below for more detailed discussion).

It is advisable to check the quality of the data covariance matrix by inspecting the rank and condition number, and also the singular value spectrum of the matrix, which is obtained by a Singular Value Decomposition (SVD). Plotting the singular values on a logarithmic axis provides an indication of the effective numerical rank of the covariance matrix. Typically, the singular value spectrum of an ill-conditioned matrix shows a “cliff” at its effective rank, with the numerically irrelevant components having singular values several orders of magnitude smaller than the rest (Fig. 1). Sometimes, the spectrum can show two or more such cliffs, e.g., in the case of combined channel types (Fig. 1C) or when the data has been processed with SSS (Fig. 1D). In such cases, a close inspection of the singular value spectrum is advisable, as conventional rank estimation via SVD can fail. In the following paragraph, we will further explain why ill-conditioned covariance matrices pose a problem for

the computation of the spatial filter, and what can be done to mitigate this problem.

**Mathematical inversion of ill-conditioned covariance matrices** If the estimate of the data covariance is unreliable, and thus proves to be ill-conditioned, a simple mathematical inverse of this matrix is either impossible, causing the beamformer computation to fail completely, or the used ill-conditioned covariance matrix will lead to poor beamformer results. To ensure a numerically stable inversion of the covariance matrix, one can use a truncated pseudo-inverse, or use “diagonal loading” as a regularization technique (Hillebrand and Barnes, 2003; 2005). Diagonal loading makes the data covariance matrix full-rank by adding a small constant to the diagonal elements of the matrix (Vrba and Robinson, 2000). This technique is also known as Tikhonov regularization (Tikhonov and Arsenin, 1977):

$$\mathbf{R}_{\text{reg}} = \mathbf{R} + \lambda \mathbf{I} \quad (19)$$

where  $\mathbf{I}$  is an  $M \times M$  identity matrix, and  $\lambda$  represents the regularization parameter. Here,  $\lambda$  needs to be defined by the researcher, and is often chosen to be a percentage of the average sensor power  $\text{Tr}(\mathbf{R})/M$  (cf. FieldTrip, MNE-Python, Brainstorm or SPM) or can be set to the lowest singular value of the covariance matrix (Robinson and Vrba, 1999; Woolrich et al., 2011).

The regularization of the covariance matrix results in a broader pass-band of the spatial filter, which increases the output SNR of the filter, but also spatially blurs the source estimates (Brookes et al., 2008). Thus, a trade-off between increasing output SNR and decreasing spatial resolution exists and makes the choice of the regularization parameter  $\lambda$  crucial for beamformer performance.

There is no easy analytic way to determine the regularization parameter in Tikhonov regularization for MEG data in order to obtain the best result with the least spatial blurring. For other regularization techniques such as Bayesian PCA regularization (Woolrich et al., 2011), the explicit choice of a regularization parameter is not needed. The Ledoit-Wolf shrinkage model (Ledoit and Wolf, 2004a) provides another way of obtaining a well conditioned covariance matrix via shrinkage without the need to choose the shrinkage parameter (Engemann and Gramfort, 2015; Ledoit and Wolf, 2004b). Yet both of these techniques have underlying assumptions on the data: temporal independence for Ledoit-Wolf, and low-rank plus sensor independent noise for PCA. The aforementioned truncated pseudo-inverse is another way to potentially bypass the choice of a regularization parameter. Since this strategy is not applied at the data covariance matrix estimation step but during the inversion process, this needs to be implemented in the beamformer algorithm (cf. Section 4).

*Spatial whitening of covariance matrices* In some situations it might be useful to spatially whiten the covariance matrix. Whitening, also called pre-whitening, is a linear operation that intends to decorrelate and scale the noise components in the data. The term whitening refers to the color of the noise being white, i.e., having a covariance matrix that equals the Identity matrix. To achieve this, the procedure uses a so-called noise covariance matrix, a channel-level covariance matrix (cf. Eq. (3)) which can be computed on an empty room recording or a pre-stimulus quiescent baseline. There are different approaches to whitening such as principal component analysis (PCA) or zero-phase component analysis (ZCA). Using an eigendecomposition on the noise covariance matrix yields  $\mathbf{R}_{\text{noise}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$  with eigenvectors  $\mathbf{U}$  and eigenvalues  $\mathbf{\Lambda}$ . The whitening matrix, that is subsequently applied to the data and the forward field, is then computed as  $\mathbf{V} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T$  in the PCA case and  $\mathbf{V} = \mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T$  in the ZCA case (cf. Engemann and Gramfort, 2015). Pre-whitening has been explicitly described in the context of beamforming as well (Sekihara et al., 2008), but has not been widely adopted as a standard pre-processing step. Note that theoretically, for many the beamformer equations to hold, e.g., as per Sekihara and Nagarajan (2008), the noise is assumed to be spatially white. In practice, spatial whitening is useful when combining different sensor types (cf. Section 3.6), because it scales the magnitude of the different sensor types relative to one another, or when dealing with severely rank-deficient data as described in the next paragraph.

*Signal space separation leads to severely rank deficient covariance matrices* The application of SSS to MEG data for cleaning (e.g., through the MEGIN software, using MNE-Python's `mne.preprocessing.maxwell_filter` function or the TSSS toolbox in SPM) often results in a severe rank-deficiency. In a 306-channel data set, it is not unusual for several tens, sometimes above 200 spatial components to be removed from the data (cf. Fig 1). In this case, regularization by means of diagonal loading is not sufficient for good beamformer performance. Using a truncated pseudo-inverse in combination with pre-whitening of the covariance matrix (and forward field) has worked well on all examples we tested and can be a good first choice. Details on if and how this can be implemented in the

different toolboxes are given in Section 4.2 (truncated pseudo-inverse) and 4.4 (whitening).

### 3.2. The role of forward field accuracy

The second ingredient to the beamformer is the forward field matrix (cf. Eq. (5)). Consequently, the quality of the forward field (also called forward model) is crucial for good beamformer performance (Brookes et al., 2008; Hillebrand and Barnes, 2003, 2011; Steinsträter et al., 2010). The intricacies of forward modelling are beyond the scope of this paper; nevertheless, some details regarding forward field accuracy will be discussed in the following.

The forward field solution is determined by the source model, the sensor or electrode configuration, and the volume conductor model. Inaccuracies in the forward field arise mainly from sources of error concerning the latter two. Especially with EEG, the accuracy of the volume conductor model is important, as the complicated volume conduction of electrical currents requires realistic head models (Hamalainen and Sarvas, 1987; Hillebrand and Barnes, 2005; Steinsträter et al., 2010; Wolters et al., 2006). To what extent realistic head models may improve the performance of MEG source modelling is debated (Huang et al., 1999; Neugebauer et al., 2017), but beamformers are generally more sensitive to errors in the volume conductor model than other source reconstruction methods (Brookes et al., 2008; Hillebrand and Barnes, 2003; Sekihara et al., 2002; Steinsträter et al., 2010).

The coregistration of the sensor space (MEG head space or EEG electrode space) with the head model space (usually MRI space) is another possible source of error in the forward model. Coregistration errors have been shown to range in the order of several millimeters (Adjamian et al., 2004; Dalal et al., 2014; Singh et al., 1997). In the case of EEG, they were demonstrated to not only impact the localization accuracy of the reconstructed source, but also the output SNR of the reconstruction, thus raising the potential of missing weak or deep sources (Dalal et al., 2014).

A more fundamental problem arises from the constitution of the source model. Since the sources are modelled as dipole sources, true extended sources can be misrepresented in the reconstruction, as the activity from neighbouring source points will be suppressed in the beamformer filter (Vrba, 2002).

Generally and perhaps counterintuitively, the impact of errors in the forward model increases with increasing SNR of the data. For high SNR data, the spatial filter becomes sharper, and the resolution therefore becomes high enough for a discrimination between the real and the modelled forward field (Cox, 1973; Godara, 1986; Hillebrand and Barnes, 2003).

### 3.3. Rank reduction of forward fields

For a dipole with unconstrained orientation, the forward field has three columns, corresponding to three orthogonal directions. With respect to EEG data, the intrinsic rank of a forward field is three, since dipoles with any orientation (i.e., radial and tangential to the head surface) can be measured. MEG measurements, however, are only weakly sensitive to sources radial to the head surface. With a spherical head model, the rank of the forward field is thus naturally reduced to 2, even though any dipole has a genuine radial component. In realistic head models (e.g., single shell or boundary element head models), the orientation sensitivity of the MEG is more variable. However, one orientation component is still close to zero at most locations, and thus can potentially carry a lot of noise. Therefore, the option to reduce the rank of the forward field is generally advised for MEG data. This is done by decomposing the forward field per location with an SVD and then back-projecting the forward field by zeroing the smallest singular value.

### 3.4. The choice of spatial distribution of source models: volumetric versus cortically constrained meshes

Beamforming estimates neural activity *independently* for each position, typically representing an individual dipolar source; this is repeated for positions distributed over the whole brain or a smaller volume of interest. It is important to note that the beamformer solution for any given location is not influenced by which other locations are included in the volume of interest. This procedure is different to dipole fitting and distributed source models (such as minimum norm estimation). In dipole fitting, only one or very few sources are obtained (*i.e.*, those that explain the sensor-level data best), and in distributed source models, the distributed source strength is estimated jointly across the source space, most typically a cortical sheet. This means that the solution at a given location is heavily dependent on which other locations are included in the source space. In beamforming, however, the sequential scanning does not impose constraints to use a specific arrangement of the individual sources: whether they are positioned along a cortical sheet or in a volumetric grid does not have consequences for the estimated beamformer output at a given source point. There is no agreement on the best practice to use either a cortical sheet or an arrangement in a regular volumetric 3-D grid, so the choice depends on the research question and the planned follow-up analyses. While cortical sheets can be reflective of MEG and EEG data properties, a volumetric mesh allows for recovering the activity from subcortical sources. A volumetric model also makes it easier to identify anatomical mis-alignment or other incorrect beamformer results, *e.g.*, a center of head bias (*cf.* Section 2.1).

### 3.5. Beamforming of EEG data

Beamforming of EEG data is possible but requires being aware of some additional considerations. Firstly, the forward field is to be calculated with the same reference as the sensor space data. An average reference is a good choice, as it mitigates the modelling error introduced by localization inaccuracies of the electrodes' positions. With a single reference electrode the model error of this electrode will be passed to all other electrodes, and this error decreases through averaging across all electrodes. Secondly, as briefly mentioned above, the accuracy of the forward field is of high importance, as EEG is very sensitive to the head conduction profile (Hamalainen and Sarvas, 1987; Hillebrand and Barnes, 2005; Steinsträter et al., 2010; Wolters et al., 2006). This necessitates the use of a realistic head model, *e.g.*, a boundary element or finite element model. This model should ideally be created from the individual MRI scan of the participant to maximize accuracy of the forward model.

### 3.6. Beamforming with combined sensor types

Some neuroscientific data sets contain signals from different sensor types, for example when EEG and MEG have been recorded simultaneously, or when an MEG system has been used that has both magnetometer and gradiometer channels. In order to take full advantage of the slightly different type of information that is provided by these different types of sensors (*e.g.*, magnetometers being more sensitive to deeper sources than gradiometers, and MEG not being sensitive to dipoles with a radial orientation), the joint use of different sensor types for beamformer source reconstruction might be of interest. This, however, requires either the re-scaling of the data or sensor type specific values for regularization, since different sensor types are measured at different scales (*e.g.*, MEG data from gradiometers are usually in the range of  $10^{-11}\text{Tm}^{-1}$  and data from magnetometers in the range of  $10^{-13}\text{T}$ ). If the data is not scaled, the beamformer solution will be biased towards the sensor type with the larger values. One convenient way to achieve a proper integration of both sensor types is by spatially whitening the forward field and the covariance matrix using an appropriately

estimated noise covariance matrix (*cf.* Section 3.1). However, this approach requires either an empty room recording or a pre-stimulus quiescent baseline, which are not always available. Alternatively, the sensor types can also simply be scaled relative to each other using a constant. Additionally, in creating the spatial filters, the cross terms of the covariance matrix, *i.e.*, all the covariance values between two different types of sensors, are typically set to zero. Both of these approaches, however, do not take into account that different sensor or electrode types can have different spatial sensitivity profiles.

### 3.7. Common choices for beamformer analysis pipelines

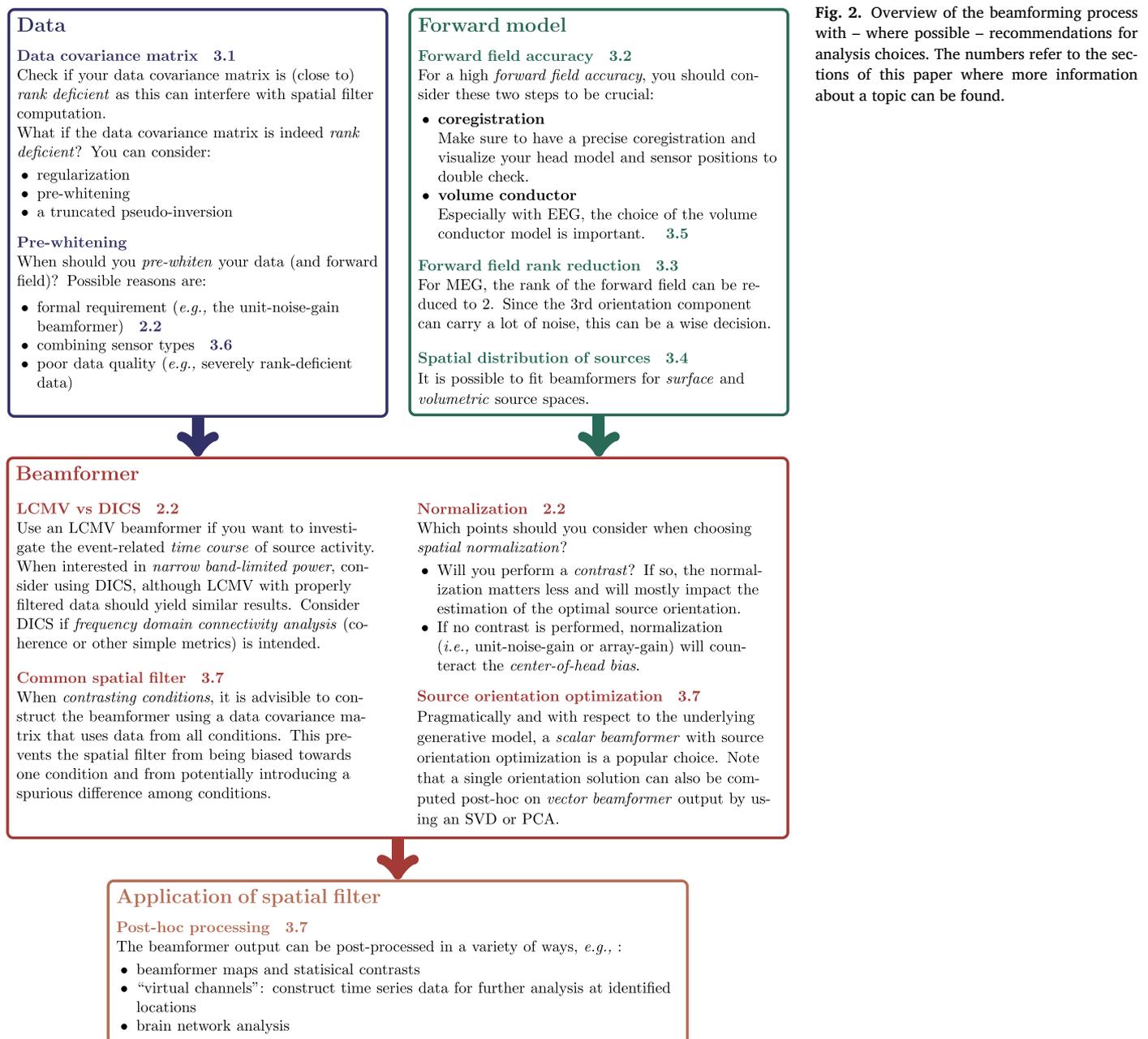
The successful estimation of source activity using beamformers relies on several experimental and data processing choices. Regarding a thorough and comprehensive review of good experimental design, we refer the reader to Brookes et al., 2008 and Gross et al. (2013). In the following, we will have a closer look at some of the most relevant points, especially with respect to the topics discussed in Section 3. For quick reference, Fig. 2 contains an overview of the beamforming process and its most critical parts.

**Experimental setup** Source reconstruction of M/EEG should already be anticipated in the experimental design. For a good estimate of the data covariance matrix, the amount of data (in terms of number of channels, signal bandwidth, and number of samples) plays an important role (Brookes et al., 2008). If it is anticipated that the data will be pre-whitened, *e.g.*, because different sensor types will be combined or SSS will be applied, an empty-room recording or quiescent baseline should be included. A pre-stimulus baseline is also important if a contrast against baseline will be computed, in which case the integration window for the covariance should be equally long for the baseline and "active" period (also see the paragraph about common spatial filters below). Especially if beamforming is planned for an EEG study, it is important to acquire individual MRIs to obtain an accurate forward model. Good care with the coregistration procedure is generally advised (Adjamian et al., 2004; Dalal et al., 2014; Gross et al., 2013).

**Beamformer parameters** LCMV beamformers can be used to estimate event-related time courses of source activity. DICS beamformers can be used to estimate source power of band-limited activity and to perform frequency domain connectivity analysis. Which beamformer type to choose (unit-gain, array-gain, or unit-noise-gain) depends on the experimental question. If no experimental contrast is computed, a spatial normalization is a sensible choice to mitigate the center of head bias (*cf.* Section 2.2). If, on the other hand, the beamformer output is used to form a contrast between *e.g.*, two conditions, the choice of beamformer type is less critical. It is important to note, however, that differences in the estimation of the optimal orientation (*e.g.*, Eqs. (7) and (15)) still play a role with the computation of contrasts. Whether or not to compute the optimal source orientation is partially an empirical question as it depends on implicit assumptions with respect to the underlying generative model as well as on pragmatic considerations. If one assumes that each source can be well described as a single current dipole, then this justifies a fixed orientation from the modelling point of view. Pragmatically, depending on how the beamformer output will be subsequently processed, a scalar beamformer output may be more practical (*e.g.*, it preserves polarity information of event-related responses). Note, that a single orientation solution can also be obtained by SVD (or PCA) from a vector beamformer output.

**Common spatial filters** If a contrast between two conditions is formed, it is good practice to compute the data covariance matrix based on an equal amount of data from both conditions (Gross et al., 2013). This *common spatial filter* is subsequently applied to each condition separately. This procedure prevents the spatial filter from being skewed and introducing spurious differences among the two conditions.

**Post-hoc processing** The beamformer output can be post-processed in a variety of ways, for which there are generally applicable recommendations to be given. Possible scenarios are to inspect the beamformer



**Fig. 2.** Overview of the beamforming process with – where possible – recommendations for analysis choices. The numbers refer to the sections of this paper where more information about a topic can be found.

maps or resulting statistical contrasts for local extrema, and to use the identified locations as “virtual channels”. The spatial filters for those locations can then be used to construct time series data which can be processed in a variety of ways (*e.g.*, single trial classification analysis, time-frequency decomposition, connectivity analysis). Another common use case is to employ the spatial filter of all locations to obtain a whole brain estimate of source time courses, which can subsequently be used for brain network analysis, possibly after dimensionality reduction using a parcellation scheme.

**Group studies** For group studies and contrasts between groups, it is advisable to keep the beamformer parameters and processing steps (*e.g.*, type of normalization, amount of regularization, *etc.*) identical across participants. Parameters can be determined based on a small amount of data sets, *e.g.*, pilot participants. Depending on the experimental design, a suited localizer task could also be used to establish appropriate parameters.

#### 4. Beamformer implementations in open source software

In this section, we compare the support for beamformer implementations in four frequently used open source M/EEG data analysis toolboxes: Brainstorm, FieldTrip, MNE-Python, and SPM (DAiSS toolbox). We present how those toolboxes implement the different beamformer variants and examine how they handle issues with beamforming source reconstruction.

The rather independent development of beamformer implementations across the toolboxes resulted in considerable differences in the user interfaces and nomenclature of different options. However, it is important to note that the actual implementations in the toolboxes all depend on the formulas discussed in this paper. Thus, the different implementations converge well in their output and all yield reliable source localization for typical signal-to-noise ratios, as has lately been shown by Jaiswal et al. (2020).

Apart from differences regarding the beamformer implementations, the toolboxes differ also on a bigger scale: while FieldTrip and MNE-Python are script-based analysis toolboxes, Brainstorm and SPM are primarily used via a graphical user interface (GUI), although they can be used with scripted analyses as well. While FieldTrip, Brainstorm and SPM rely on MATLAB, MNE-Python is written in Python. Part of these differences can be attributed to the historic development of the toolboxes, influenced in part by the available hardware and software as well as the specific research questions of the authors and maintainers of these toolboxes.

#### 4.1. Implemented beamformer variants

The following section gives an overview of the different function calls across the toolboxes to compute spatial filters. In addition, we refer the reader to Tables S1–S4 in the Supplementary Material, which show how to produce a unit-gain (Table S1), array-gain (Table S2) or unit-noise-gain LCMV beamformer (Table S3) or a DICS beamformer (Table S4). The Supplementary Material also contains a list with links to tutorials and documentation of best practices in beamforming for all four toolboxes.

In **Brainstorm**, `process_source_2018` (as used in Jaiswal et al., 2020) implements the LCMV beamformer. The forward model is computed on either a cortical surface sheet, with constrained (scalar) or unconstrained (vector) source models, or it is computed on an unconstrained (vector) volume source grid. The data covariance is used to “data-whiten” the forward model, with automatic adaption for averaged vs. continuous data sets, note that this differs from the procedure described in Section 3.6, where a specific noise covariance is used. A form analogous to the array-gain method of the beamformer is computed in this data-whitened space, so that the image is unitless and can be viewed as a “z-scored” image. The function `process_extract_scouts` extracts either scalar and vector time series; as discussed in 2.2, only the norm of the vector time series is recommended. A post-processing step, `process_dipole_scanning`, extracts from these time-series the optimal dipole location and orientation at each time step, in proper units of A m. Brainstorm does not include a DICS implementation, but an extension exists which calls FieldTrip functions from within Brainstorm (by V. Youssofzadeh, <https://github.com/vyoussofzadeh/DICS-beamformer-for-Brainstorm>).

In **FieldTrip**, a beamformer is calculated and applied to the sensor data covariance in a single step, using the function `ft_sourceanalysis`. This function takes a precomputed sensor data covariance matrix or CSD matrix and a configuration structure as input. The configuration structure `cfg` contains the settings, but also the forward model. Specifying `cfg.method = 'lcmv'` computes an LCMV beamformer, while `cfg.method = 'dics'` computes a DICS beamformer. All other options and beamformer specifications are also specified in this configuration structure (cf. Tables S1–S4). If data whitening is required, this can be done prior to the computation of the sensor covariance and the forward model using the function `ft_denoise_prewhiten`.

In **MNE-Python**, the computation and application of the spatial filter are separated. The LCMV beamformer weights are computed using the function `make_lcmv()`, which relies on the measurement information, the data covariance, and a precomputed forward model. All other parameters are specified in the function call (cf. Tables S1–S4). The obtained filter can then be applied to different data: to averaged data using `apply_lcmv()`, to a data covariance using `apply_lcmv_cov()`, to epochs using `apply_lcmv_epochs()`, and to raw data using `apply_lcmv_raw()`. The DICS beamformer uses a similar user interface to the LCMV beamformer, and can be estimated through `make_dics()` and applied via a range of `apply_dics()` functions.

In **DAiSS (SPM)**, it is possible to build pipelines by selecting different options for a fixed sequence of modules. The forward model computation is done in the `Define sources` module and the co-

variance (for LCMV) or CSD (for DICS) matrix is computed in the `Covariance features` module where also different options for regularization can be specified. The choice between LCMV and DICS is made in the `Inverse solution` module.

#### 4.2. Regularization and inversion of covariance matrices

**Brainstorm** estimates the empirical noise and data covariance matrices from the user’s selection of baseline and data segments. Five variations of regularization are offered: `regularize`, which is similar to the methods described in Section 3.1, where a fraction of the average eigenvalue, e.g., 0.1, is added to the eigenvalues; `median`, where the median eigenvalue is repeated for all of the smaller eigenvalues; `diagonal` where only the diagonal covariances are retained; `none` where no regularization is performed; and `shrinkage` which applies the Ledoit-Wolf method of regularization. Prior to applying any of these, the covariance matrix is first tested for rank deficiency at the single floating point precision and smaller eigenvalues are removed.

**FieldTrip** estimates the empirical covariance matrix. The data covariance matrix can be regularized with a defined amount of the sensor power by setting `cfg.lambda` in `ft_sourceanalysis`, or using a percentage value, e.g., ‘5%’. This will add 5% of the average sensor power to the covariance matrix diagonal for regularization. Per default, no regularization is applied. Alternatively, for a truncated pseudo-inverse, the option `cfg.kappa` needs to be specified, as the number at which the singular value spectrum of the covariance matrix will be truncated.

**MNE-Python** offers various ways to estimate the covariance matrix. Examples include the empirical covariance matrix or a Ledoit-Wolf estimator. Furthermore, the possibility to find the best estimator based on cross-validation is implemented (Engemann and Gramfort, 2015). Regularization is applied by setting the parameter `reg` in `make_lcmv()` or `make_dics()`. The default for both functions is 0.05, corresponding to a regularization with 5% of the average sensor power. If the data is rank-deficient, MNE-Python automatically computes a truncated pseudo-inverse with the possibility to manually set the rank for the truncation.

**DAiSS (SPM)** has several options for regularization with the most commonly used being `User-specified regularisation` (specified as a percentage, the same as in FieldTrip) and `Manual truncation` which reduces the dimensionality of the data prior to beamforming. For rank-deficient data, it is recommended to use the latter option and set it to be below the first “cliff” in the eigenvalue spectrum (see Fig. 1).

#### 4.3. Forward field rank reduction

**Brainstorm**’s default for forward model rank reduction is automatically applied as part of the source modeling procedure, testing each forward field for its rank at “single” floating-point precision and removing any smaller components.

**FieldTrip**’s default for rank reduction depends on the input data: while for EEG data the rank of the leadfield is not reduced, for MEG data the rank is reduced from 3 to 2. The default behaviour can be changed by setting `cfg.reducerank` to either ‘yes’ (rank will be reduced by 1) or ‘no’ when calling `ft_sourceanalysis`. A rank-reduced forward field will by default be back-projected into the original space while omitting one direction, meaning it will still have 3 columns. This can be changed by setting `cfg.backproject` to ‘no’.

**MNE-Python** does not reduce the rank of the forward field by default. This behaviour can be changed by supplying `reduce_rank=True` to the function call, which will reduce the rank by 1. The forward field will always be back-projected.

In **DAiSS (SPM)**, the rank reduction is specified in the `Define sources` module as `Reduce rank`. By default the rank is set to 2 for MEG and to 3 for EEG.

#### 4.4. Combining sensor types and pre-whitening

**Brainstorm** allows multiple sensor types, but cautions the user about mixing EEG and MEG. For all sensor types, the cross-covariances between sensor types are defaulted to be forced to zero; however, the user can override `CROSS_COVARIANCE_CHANNELTYPES=false` in the code to be `true` if they want to test using the full cross covariances. The noise covariance matrix is then used to pre-whiten the forward model and data to balance the disparity in units, similar to MNE-Python, for a combined model.

**FieldTrip** has a dedicated function for whitening (`ft_denoise_prewhiten`), which can be used when sensor types are to be combined. The whitening matrix is applied to both the numeric data, and added as a balancing scheme to the sensor array. Optionally, the cross-terms between different sensor types can be set to zero, prior to the computation of the whitening matrix with the option `cfg.split`. When using whitened data for the covariance computation and subsequent analysis, the forward model needs to be computed from the balanced sensor array.

**MNE-Python** combines different sensor or channel types through pre-whitening of the covariance matrix, leadfield, and data. Furthermore, the cross-terms between EEG and MEG channels in the covariance matrix are set to zero.

In **DAISS (SPM)**, there is an option to fuse sensor types in the **Covariance features** module. It is assumed that the sensor signals are already properly scaled prior to running DAISS. One way this can be done is by using the **Whitening** option in the **Reduce** function (`spm_eeg_reduce`). As in **FieldTrip**, the whitening matrix in this case is applied to both the data and the sensors. There are options to set the cross-terms to zero both when whitening and when computing covariance matrices for beamforming.

#### 5. Conclusion and outlook

In this technical note we provided a concise overview of the beamformers that are most commonly used for source reconstruction of EEG and MEG data. In addition, we detailed some practical considerations and best practices that need to be taken into account when employing beamformers. Finally, we compared the settings that are needed to perform the described beamformer analyses for four commonly used open source software toolboxes. This work complements earlier work (Jaiswal et al., 2020), which compared and discussed the behavior of the beamformers as applied with the respective toolboxes' default options. Here, we show that all toolboxes have sufficient functional flexibility in order to accommodate the different beamformers we described. In addition, most toolboxes provide flexibility with respect to estimation and regularization of covariance matrices, and with respect to data whitening. The implementational differences between the toolboxes result from the fact that they have developed organically over the years, all with a slightly different focus and audience. Yet, as can be also witnessed from this paper, the main developers all strive to advance the field as a whole, and regularly interact with each other to share ideas, implementational details, and code. These interactions have greatly contributed to the overall quality and functionality of the toolboxes and in the future will continue to provide the neuroscience community with the necessary means to get the most out of their data.

#### Data availability

The data that were used to generate the illustrative figure are openly available through MNE-Python. The openly available code (see below) demonstrates how to download this data set.

#### Code availability

The code used to generate the figure of this study is available under [https://github.com/britta-wstnr/beamformer\\_examples](https://github.com/britta-wstnr/beamformer_examples).

#### Credit authorship contribution statement

**Britta U. Westner:** Conceptualization, Formal analysis, Visualization, Project administration, Software, Writing – original draft, Writing – review & editing. **Sarang S. Dalal:** Supervision, Writing – review & editing, Resources. **Alexandre Gramfort:** Software, Writing – review & editing. **Vladimir Litvak:** Software, Writing – review & editing. **John C. Mosher:** Software, Writing – review & editing. **Robert Oostenveld:** Conceptualization, Software, Writing – review & editing, Project administration, Funding acquisition. **Jan-Mathijs Schoffelen:** Supervision, Software, Writing – original draft, Writing – review & editing.

#### Acknowledgment

We thank Caroline Witton, Jukka Nenonen, Amit Jaiswal, and Lauri Parkkonen for useful discussions.

This work was supported by the Innovative Training Network, Child-Brain, funded by the Marie Curie Actions of the European Commission (H2020-MSCA-ITN-2014). The Wellcome Centre for Human Neuroimaging is supported by core funding from Wellcome [203147/Z/16/Z]. The UK MEG community is supported by the MRC UKMEG Partnership grant MR/K005464/1. This work was supported by the European Union's Horizon 2020 Marie Skłodowska Curie Individual Fellowship (grant agreement 893912) to BUW. This work was supported by an ERC Starting Grant (ERC-StG-640448) to SSD. This work was supported by the ERC Starting Grant SLAB ERC-StG-676943 and the French ANR-19-DATA-0023 to AG. This work was supported by The Netherlands Organisation for Scientific Research (NWO Vidi: 864.14.011) to JMS. Research was supported in part by the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health under award numbers R01EB026299 and U01EB023820 to JCM. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

#### Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.neuroimage.2021.118789](https://doi.org/10.1016/j.neuroimage.2021.118789).

#### References

- Adjamian, P., Barnes, G.R., Hillebrand, A., Holliday, I.E., Singh, K.D., Furlong, P.L., Harrington, E., Barclay, C., Route, P., 2004. Co-registration of magnetoencephalography with magnetic resonance imaging using bite-bar-based fiducials and surface-matching. *Clin. Neurophysiol.* 115, 691–698.
- Backus, A.R., Schoffelen, J.M., Szabenyi, S., Hanslmayr, S., Doeller, C.F., 2016. Hippocampal-prefrontal theta oscillations support memory integration. *Curr. Biol.* 26, 450–457.
- Baillet, S., Mosher, J.C., Leahy, R.M., 2001. Electromagnetic brain mapping. *IEEE Signal Process. Mag.* 18, 14–30.
- Borgiotti, G., Kaplan, L., 1979. Superresolution of uncorrelated interference sources by using adaptive array techniques. *IEEE Trans. Antennas Propag.* 27, 842–845.
- Brookes, M.J., Vrba, J., Robinson, S.E., Stevenson, C.M., Peters, A.M., Barnes, G.R., Hillebrand, A., Morris, P.G., 2008. Optimising experimental design for MEG beamformer imaging. *NeuroImage* 39, 1788–1802.
- Buzsaki, G., 2006. *Rhythms of the Brain*. Oxford University Press.
- Cox, H., 1973. Resolving power and sensitivity to mismatch of optimum array processors. *J. Acoust. Soc. Am.* 54, 771–785.
- Dalal, S., Zumer, J., Agrawal, V., Hild, K., Sekihara, K., Nagarajan, S., 2004. NUTMEG: as neuromagnetic source reconstruction toolbox. *Neurol. Clin. Neurophysiol. NCN* 2004, 52.
- Dalal, S.S., Guggisberg, A.G., Edwards, E., Sekihara, K., Findlay, A.M., Canolty, R.T., Berger, M.S., Knight, R.T., Barbaro, N.M., Kirsch, H.E., Nagarajan, S.S., 2008. Five-dimensional neuroimaging: localization of the time-frequency dynamics of cortical activity. *NeuroImage* 40, 1686–1700.
- Dalal, S.S., Rampp, S., Willomitzer, F., Ettl, S., 2014. Consequences of EEG electrode position error on ultimate beamformer source reconstruction performance. *Front. Neurosci.* 8.
- Dalal, S. S., Zumer, J. M., Guggisberg, A. G., Trumpis, M., Wong, D. D. E., Sekihara, K., Nagarajan, S. S., 2011. MEG/EEG source reconstruction, statistical evaluation, and visualization with NUTMEG. <https://www.hindawi.com/journals/cin/2011/758973/abs/>.
- Dale, A.M., Sereno, M.I., 1993. Improved localization of cortical activity by combining EEG and MEG with MRI cortical surface reconstruction: a linear approach. *J. Cogn. Neurosci.* 5, 162–176.

- van Drongelen, W., Yuchtman, M., Van Veen, B.D., van Huffelen, A.C., 1996. A spatial filtering technique to detect and localize multiple sources in the brain. *Brain Topogr.* 9, 39–49.
- Engemann, D.A., Gramfort, A., 2015. Automated model selection in covariance estimation and spatial whitening of MEG and EEG signals. *Neuroimage* 108, 328–342.
- Godara, L.C., 1986. Error analysis of the optimal antenna array processors. *IEEE Trans. Aerosp. Electron. Syst.* AES-22, 395–409.
- Gramfort, A., Luessi, M., Larson, E., Engemann, D.A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., Hämäläinen, M., 2013. MEG and EEG data analysis with MNE-Python. *Front. Neurosci.* 7.
- Gramfort, A., Luessi, M., Larson, E., Engemann, D.A., Strohmeier, D., Brodbeck, C., Parkkonen, L., Hämäläinen, M.S., 2014. MNE software for processing MEG and EEG data. *Neuroimage* 86, 446–460.
- Gross, J., Baillet, S., Barnes, G.R., Henson, R.N., Hillebrand, A., Jensen, O., Jerbi, K., Litvak, V., Maess, B., Oostenveld, R., Parkkonen, L., Taylor, J.R., van Wassenhove, V., Wibral, M., Schoffelen, J.M., 2013. Good practice for conducting and reporting MEG research. *Neuroimage* 65, 349–363.
- Gross, J., Kujala, J., Hämäläinen, M., Timmermann, L., Schnitzler, A., Salmelin, R., 2001. Dynamic imaging of coherent sources: studying neural interactions in the human brain. *Proc. Natl. Acad. Sci.* 98, 694–699.
- Hämäläinen, M.S., Ilmoniemi, R.J., 1994. Interpreting magnetic fields of the brain: minimum norm estimates. *Med. Biol. Eng. Comput.* 32, 35–42.
- Hamalainen, M.S., Sarvas, J., 1987. Feasibility of the homogeneous head model in the interpretation of neuromagnetic fields. *Phys. Med. Biol.* 32, 91–97.
- Hillebrand, A., Barnes, G.R., 2003. The use of anatomical constraints with MEG beamformers. *Neuroimage* 20, 2302–2313.
- Hillebrand, A., Barnes, G.R., 2005. Beamformer analysis of MEG data. In: *International Review of Neurobiology*, Vol. 68. Elsevier, pp. 149–171.
- Hillebrand, A., Barnes, G.R., 2011. Practical constraints on estimation of source extent with MEG beamformers. *Neuroimage* 54, 2732–2740.
- Huang, M.X., Mosher, J.C., Leahy, R.M., 1999. A sensor-weighted overlapping-sphere head model and exhaustive head model comparison for MEG. *Phys. Med. Biol.* 44, 423–440.
- Jaiswal, A., Nenonen, J., Stenroos, M., Gramfort, A., Dalal, S.S., Westner, B.U., Litvak, V., Mosher, J.C., Schoffelen, J.M., Witton, C., Oostenveld, R., Parkkonen, L., 2020. Comparison of beamformer implementations for MEG source localization. *Neuroimage* 216, 116797.
- Lalancette, M., 2014. Rotational invariance and source orientation in LCMV Vector beamformer. In: *Poster at the 19th International Conference on Biomagnetism (Halifax)*.
- Ledoit, O., Wolf, M., 2004a. Honey, I shrunk the sample covariance matrix. *J. Portf. Manag.* 30, 110–119.
- Ledoit, O., Wolf, M., 2004b. A well-conditioned estimator for large-dimensional covariance matrices. *J. Multivar. Anal.* 88, 365–411.
- Litvak, V., Mattout, J., Kiebel, S., Phillips, C., Henson, R., Kilner, J., Barnes, G., Oostenveld, R., Daunizeau, J., Flandin, G., Penny, W., Friston, K., 2011. EEG and MEG data analysis in SPM8. <https://www.hindawi.com/journals/cin/2011/852961/>.
- Litvak, V., Eusebio, A., Jha, A., Oostenveld, R., Barnes, G.R., Penny, W.D., Zrinzo, L., Hariz, M.I., Limousin, P., Friston, K.J., Brown, P., 2010. Optimized beamforming for simultaneous MEG and intracranial local field potential recordings in deep brain stimulation patients. *Neuroimage* 50, 1578–1588.
- Mosher, J.C., Baillet, S., Darvas, F., Pantazis, D., Kucukaltun-Yildirim, E., Leahy, R.M., 2005. Brainstorm electromagnetic imaging software. *Int. J. Bioelectromagn.* 7, 2.
- Neugebauer, F., Möddel, G., Rampp, S., Burger, M., Wolters, C.H., 2017. The effect of head model simplification on beamformer source localization. *Front. Neurosci.* 11.
- Oostenveld, R., Fries, P., Maris, E., Schoffelen, J.M., 2011. FieldTrip: open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Intell. Neurosci.* 2011, 1:1–1:9.
- Quraan, M.A., Moses, S.N., Hung, Y., Mills, T., Taylor, M.J., 2011. Detection and localization of hippocampal activity using beamformers with MEG: a detailed investigation using simulations and empirical data. *Hum. Brain Mapp.* 32, 812–827.
- Robinson, S.E., Vrba, J., 1999. Functional neuroimaging by synthetic aperture magnetometry. In: *oshimoto, T., Kotani, M., Kuriki, S., Karibe, H., Nakasato, N. (Eds.), Recent Advances in Biomagnetism*, pp. 302–305.
- Rousseeuw, P.J., Driessen, K.V., 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41, 212–223.
- Schoffelen, J.M., Oostenveld, R., Fries, P., 2005. Neuronal coherence as a mechanism of effective corticospinal interaction. *Science* 308, 111–113.
- Sekihara, K., Hild, K.E., Dalal, S.S., Nagarajan, S.S., 2008. Performance of prewhitening beamforming in MEG dual experimental conditions. *IEEE Trans. Biomed. Eng.* 55, 1112–1121.
- Sekihara, K., Nagarajan, S.S., 2008. *Adaptive Spatial Filters for Electromagnetic Brain Imaging*. Springer Science & Business Media.
- Sekihara, K., Nagarajan, S.S., 2015. *Electromagnetic Brain Imaging: a Bayesian Perspective*. Springer International Publishing.
- Sekihara, K., Nagarajan, S.S., Poeppel, D., Marantz, A., 2004. Performance of an MEG adaptive-beamformer source reconstruction technique in the presence of additive low-rank interference. *IEEE Trans. Biomed. Eng.* 51, 90–99.
- Sekihara, K., Nagarajan, S.S., Poeppel, D., Marantz, A., Miyashita, Y., 2002. Application of an MEG eigenspace beamformer to reconstructing spatio-temporal activities of neural sources. *Hum. Brain Mapp.* 15, 199–215.
- Singh, K.D., 2012. Which “neural activity” do you mean? fMRI, MEG, oscillations and neurotransmitters. *Neuroimage* 62, 1121–1130.
- Singh, K.D., Holliday, I.E., Furlong, P.L., Harding, G.F.A., 1997. Evaluation of MRI-MEG/EEG co-registration strategies using Monte Carlo simulation. *Electroencephalogr. Clin. Neurophysiol.* 102, 81–85.
- Steinsträter, O., Sillekens, S., Junghefer, M., Burger, M., Wolters, C.H., 2010. Sensitivity of beamformer source analysis to deficiencies in forward modeling. *Hum. Brain Mapp.* 31, 1907–1927.
- Tadel, F., Baillet, S., Mosher, J.C., Pantazis, D., Leahy, R.M., 2011. Brainstorm: a user-friendly application for MEG/EEG analysis. *Intell. Neurosci.* 2011, 8:1–8:13.
- Taulu, S., Kajola, M., Simola, J., 2003. Suppression of interference and artifacts by the signal space separation method. *Brain Topogr.* 16, 269–275.
- Tikhonov, A.N., Arsenin, V.I., 1977. *Solutions of ill-posed problems*. John Wiley, New York.
- Tipping, M.E., Bishop, C.M., 1999. Probabilistic principal component analysis. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* 61, 611–622.
- Van Veen, B., Buckley, K., 1988. Beamforming: a versatile approach to spatial filtering. *IEEE ASSP Mag.* 5, 4–24.
- Van Veen, B.D., Van Drongelen, W., Yuchtman, M., Suzuki, A., 1997. Localization of brain electrical activity via linearly constrained minimum variance spatial filtering. *IEEE Trans. Biomed. Eng.* 44, 867–880.
- Vrba, J., 2002. Magnetoencephalography: the art of finding a needle in a haystack. *Phys. C* 368, 1–9.
- Vrba, J., Robinson, S.E., 2000. Linearly constrained minimum variance beamformers, synthetic aperture magnetometry, and MUSIC in MEG applications. In: *Proceedings of the Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat. No. 00CH37154)*, Vol. 1, pp. 313–317.
- Wilson, T.W., Slason, E., Asherin, R., Kronberg, E., Reite, M.L., Teale, P.D., Rojas, D.C., 2010. An extended motor network generates beta and gamma oscillatory perturbations during development. *Brain Cogn.* 73, 75–84.
- Wolters, C., Anwander, A., Tricoche, X., Weinstein, D., Koch, M., MacLeod, R., 2006. Influence of tissue conductivity anisotropy on EEG/MEG field and return current computation in a realistic head model: a simulation and visualization study using high-resolution finite element modeling. *Neuroimage* 30, 813–826.
- Woolrich, M., Hunt, L., Groves, A., Barnes, G., 2011. MEG beamforming using Bayesian PCA for adaptive data covariance matrix regularization. *Neuroimage* 57, 1466–1479.