Noninvasive fetal QRS detection using echo state network and dynamic programming

Mantas Lukoševičius, Vaidotas Marozas

Biomedical Engineering Institute, Kaunas University of Technology,
Studentu g. 65, LT-51369 Kaunas, Lithuania

E-mail: mantas.lukosevicius@ktu.lt

Abstract. We address a classical fetal QRS detection problem from abdominal ECG recordings with a data-driven statistical machine learning approach. Our goal is to have a powerful yet conceptually clean solution. There are two novel key components at the heart of our approach: an echo state recurrent neural network that is trained to indicate fetal QRS complexes, and several increasingly sophisticated versions of statistics-based dynamic programming algorithms, that are derived from and rooted in probability theory. We also employ a standard techniques for preprocessing and removing maternal ECG complexes to the signals, but do not take this as the main focus of this work. The proposed approach is quite generic and can be extended to other type of signals and annotations. Open source code is provided.

Keywords: Abdominal ECG, Fetal QRS, Neural Network, Reservoir Computing, Echo State Network, Dynamic Programming, Statistical Machine Learning
Submitted to: *Physiol. Meas.*

1. Introduction

Monitoring of fetal ECG (fECG) and its parameters would provide important information about the fetal heart status and various distress factors. The problem, however, is difficult. Noninvasive fECG has low signal-to-noise ratio, is contaminated by the strong interferences: maternal ECG (mECG), fetal brain activity, myographic signals, movement artifacts. Forty years of research provided little to clinically significant advances in prenatal fetal ECG monitoring, see (Sameni & Clifford 2010) and the editorial in this issue (Clifford, Silva, Behar & Moody 2014) for good reviews of many existing approaches to fECG extraction and fetal heart rate (fHR) estimation. PhysioNet portal with its Challenge'2013 (Silva, Behar, Sameni, Zhu, Oster, Clifford & Moody 2013) took initiative to inspire researchers to turn to this old problem again with new methods and tools and try to move the field forward.

Matched (Farvet 1968) and adaptive filtering (Widrow, Glover, McCool, Kaunitz, Williams, Hearn, Zeidler, Eugene Dong & Goodlin 1975) were the first digital signal processing methods applied to the problem of fECG estimation and fetal heart rate extraction. Now three main classes of approaches are used to solve the problems of fECG estimation and fetal heart rate extraction: adaptive filtering, blind source separation, and ad-hoc (hand-crafted) mixture of various methods. All of them have advantages and disadvantages. The main disadvantage of adaptive filtering is the requirement for two kinds of signals: abdominal signal with mixture of fetal and mother ECGs and an ECG signal from mother's chest. The blind source separation methods (principal component analysis (Kanjilal, Palit & Saha 1997) and independent component analysis (Lathauwer, Moor, Vandewalle, Spain, Lathauwer, Callaerts & Moor 1994)) relay on the assumption that signal sources – fECG, mECG, and noise – are mixed with a linear stationary mixing matrix. If this assumption is violated, the source separation results are not adequate. They also require additional mechanisms to ensure that correct signals are taken among the blindly separated.

The authors of the summary of Physionet challenge "Noninvasive Fetal ECG: the PhysioNet/Computing in Cardiology Challenge 2013" (Silva et al. 2013) conclude that most of successful fECG detection solutions made use of fusion of several approaches e.g. (Behar, Oster & Clifford 2013) proposed and successfully used the algorithm (FUSE method) which selects the best among 4 different channels for source separation. Each channel itself is based on template subtraction, blind source separation (ICA) or their combinations.

The most successful participants of the Challenge (Andreotti, Riedl, Himmelsbach, Wedekind, Zaunseder, Wessel & Malberg 2013) also took the strategy of complex branching of the algorithm and combining many signal processing approaches: ICA for maternal signal enhancement, matched filter detector, extended Kalman smoother, template adaptation, statistical decision making. Although showing good results, later algorithms are very ad-hoc, dedicated to this particular problem of fetal heart rate estimation from particular set of signals.

We propose here a new approach for fetal QRS (fQRS) detection and heart rate estimation based on supervised machine learning. It employs two innovative key components: (i) an "echo state" recurrent artificial neural network (ESN) is trained to recognize fQRS, and (ii) several options of dynamic programming (DP) approaches are used to fuse information coming from sensors with estimated statistics of fQRS to find the most likely sequence of fQRS timings. The preliminary results of this work were presented in a shorter conference publication (Lukoševičius & Marozas 2013).

Artificial neural networks (ANNs) are powerful tools which were also used for the problem of fetal ECG extraction before, but mostly in the setting of adaptive filtering. One study (Camps-Valls, Martinez-Sober, Soria-Olivas, Magdalena-Benedito, Calpe-Maravilla & Guerrero-Martinez 2004) has found that nonlinear FIR based neural network adaptive filter clearly outperformed linear LMS based algorithm. ESN as an adaptive filter for ECG processing was introduced in (Petrėnas, Marozas, S ornmo

& Lukoševičius 2012) for QRST cancellation during atrial fibrillation. Recent study (Behar, Johnson, Clifford & Oster 2014) compared several types of linear adaptive filters (LMS, RLS) with nonlinear based on ESN for the fECG extraction task. This time, the nonlinear ESN based adaptive filter showed only slightly superior performance with respect to the LMS, RLS and template subtraction methods.

Our approach, in contrast, does not require a reference mECG lead and there is no adaptive filtering involved: the ESN is only trained once on a provided annotated data as an example and then is used fixed. To the best of our knowledge there is also no previous approach to use a probabilistic interpretation and dynamic programming to maximize the likelihood of the detected QRS annotations the way we do it.

Figure 1 outlines the components and signal flow of our approach. Apart from repairing, filtering, and normalizing the data discussed in Section 2.2, our method consists of three major steps: canceling the mECG from the signals discussed in Section 2.3, an echo state neural network discussed in Section 3.1 producing an indicator signal, and probabilistic dynamic programming algorithms detecting fetal R waves in it discussed in Section 3.2, elaborating more on the latter.

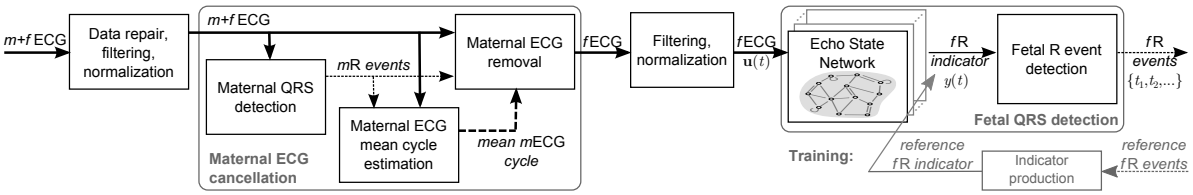

Figure 1. The block diagram of our approach. Here the bold arrows denote the four-lead ECG signals, and dotted lines denote signals that are not present for every time step (for abbreviations see the text below).

2. Signal preprocessing and mECG removal

2.1. Dataset

PhysioNet (Silva et al. 2013) provided a collection of one-minute fQRS recordings. Each recording contains four noninvasive abdominal leads. Mother chest leads were not provided. Though sampling frequency is the same 1000 Hz for all recordings, the instrumentation varied and had differing frequency response, resolution, and configuration. The data have been divided into three datasets. Dataset A (75 records) was a training set which included noninvasive fECG signals, as well as reference annotations marking the locations of each fQRS complex. Dataset B (100 records) included noninvasive fECG signals only and was used for evaluation of the Challenge entries by the organizers. The last Dataset C was reserved for evaluation of open-source Challenge entries and remained secret. The Challenge was to produce a set of annotations (fQRS complex locations) that matches the non-disclosed references as nearly as possible. fQRS complex locations are annotated by marking the R waves.

2.2. Data preprocessing

Our preprocessing first repairs the irregularities of the provided data, such as values dropping beyond the range of analog-digital converter or large initial transients (see (Lukoševičius & Marozas 2013) for more details). We then filter the signal using a bandpass filter leaving only the frequencies between 3 and 48 Hz. The resulting signal is then normalized to have zero mean and unit standard deviation. All the preprocessing discussed above is applied to every of the four leads of the ECG signal where both maternal and fetal ECGs are present.

2.3. Maternal ECG removal

The *m*ECG removal is inspired by (Martens, Rabotti, Mischi & Sluijter 2007). The implementation provided by the Challenge organizers was taken and improved upon in several respects. As depicted in Figure 1, the *m*ECG removal in turn consists of several steps.

First the *m*ECG complexes are detected on one of the four leads. We observe that the lead where *m*ECG is most pronounced, typically has the most asymmetric signal value distribution around the zero mean. For this, in every lead we compute the third statistical moment, called *skewness*, of the signal values, which has been used before, e.g., in (Behar, Oster, Li & Clifford 2013). The lead with maximal absolute skewness is selected for *m*QRS detection. In the process, all the leads are made to have a positive skewness, flipping the sign of the signal if necessary. This is a heuristic with a goal to make all *m*ECG R peaks point upwards and thus to some extent give a common format to the differently recorded signals.

The maternal R peaks are detected by finding maximal values within reasonable intervals of maternal R-R durations, as explained in more detail in Section 3.2.2.

The mean *m*ECG cycle is computed by aligning all the *m*ECG cycles by the detected R points and averaging them in all the four leads separately. The mean *m*ECG is then subtracted around every R wave from the original signal. The subtraction is done following (Martens et al. 2007): the mean ECG cycle is divided into three parts that roughly correspond to the P wave, the QRS complex, and the T wave; and for every of the three parts an optimal scaling is determined before the subtraction. Each optimal scaling is found by minimizing the square distance between the real signal at the interval of removal and the scaled averaged fragment which is to be removed. In addition to accuracy, this scaling greatly improves robustness of the maternal ECG cancellation procedure: false maternal R peak detections result in low scaling coefficients and do not disrupt the signals much.

The remaining signal is filtered again with a bandpass filter leaving only the frequencies between 9.5 and 48 Hz. These values were found through parameter tuning explained in Section 3.3, and the lower bound 9.5 Hz appears to be a compromise between removing enough of remaining *m*ECG residues, and leaving enough of *f*ECG for an effective *f*QRS detection. This value might be smaller with a better *m*ECG removal.

Upon visual inspection, this approach to canceling maternal ECG worked reasonably well, even though not perfectly in most of the signals. As a downside, it left visible mECG residues when there was a bigger variation in mECG shape during the signal. We did not invest the largest effort in perfecting the mECG removal, but concentrated on the techniques after the removal, constituting the main original contributions of this work.

3. Fetal QRS detection

We approach the fetal QRS, or more precisely fetal R (fR), detection in the preprocessed signal as a supervised machine learning task. The procedure consists of two stages. The first stage, detailed in Section 3.1, is an artificial recurrent neural network of the type Echo State Network (ESN) (Jaeger 2001, Jaeger 2007, Lukoševičius & Jaeger 2009). It gets the preprocessed signals with mECG canceled as its input, and is trained to produce a signal that indicates the fR peaks. The target signal for the training is produced from the provided Dataset A annotations. The second stage, detailed in Section 3.2, takes the non-perfect continuous fR indicator signal from the trained ESN and interprets it in a probabilistic fashion to produce the discrete fR annotations, making use of the statistics observed in the fR annotations provided in Dataset A. We discuss and test several alternatives for the second stage. Then, in Section 3.3 we discuss how we validated our approach and tuned its parameters.

3.1. Echo State Network

We use an Echo State Network (ESN) (Jaeger 2001, Jaeger 2007, Lukoševičius & Jaeger 2009) which is an artificial recurrent neural network with an update equation

$$\tilde{\mathbf{x}}(t) = \tanh\left(\mathbf{W}^{\text{in}}[1; \mathbf{u}(t)] + \mathbf{W}\mathbf{x}(t-1)\right), \tag{1}$$

$$\mathbf{x}(t) = (1-\alpha)\mathbf{x}(t-1) + \alpha\tilde{\mathbf{x}}(t), \tag{2}$$

where $\mathbf{x}(t) \in \mathbb{R}^{N_\text{x}}$ is a vector of ESN "reservoir" neuron activations and $\tilde{\mathbf{x}}(t) \in \mathbb{R}^{N_\text{x}}$ is its update, $\mathbf{u}(t) \in \mathbb{R}^{N_\text{u}}$ is the input signal, all at discrete time t, sampled at 1 kHz, $\tanh(\cdot)$ is a hyperbolic tangent neuron activation function applied element-wise, $[\cdot;\cdot]$ stands for a vertical vector concatenation, $\mathbf{W}^{\text{in}} \in \mathbb{R}^{N_\text{x}\times(1+N_\text{u})}$ and $\mathbf{W} \in \mathbb{R}^{N_\text{x}\times N_\text{x}}$ are the input and recurrent weight matrices respectively, and α is the leaking rate of the network update. This is a rather standard ESN configuration motivated in (Lukoševičius 2012). In this task $\mathbf{u}(t)$ is the preprocessed and mECG-removed signal with $N_\text{u} = 4$, and t is running from 1 to $T = 60\,000$ ms. The weight matrices $\mathbf{W}^{\text{in}}$ and $\mathbf{W}$ are generated randomly according to some simple rules and parameters (Lukoševičius 2012) described in Section 3.3.

The readout from the ESN is

$$\mathbf{y}(t) = \mathbf{W}^{\text{out}}[1; \mathbf{u}(t); \mathbf{x}(t)], \tag{3}$$

where $\mathbf{y}(t) \in \mathbb{R}^{N_\mathrm{y}}$ is the network output, $\mathbf{W}^\mathrm{out} \in \mathbb{R}^{N_\mathrm{y} \times (1+N_\mathrm{u}+N_\mathrm{x})}$ the output weight matrix, and $[\cdot;\cdot;\cdot]$ again stands for a vertical vector concatenation. In this task $\mathbf{y}(t) = y(t)$ is the fR indicator signal with $N_\mathrm{y} = 1$.

$\mathbf{W}^\mathrm{out}$ is trained using linear regression (Lukoševičius 2012), which is standard in ESNs, on the Dataset A to produce the fR indicator signal $y(t)$. The target signal $y^\mathrm{target}(t)$ for this supervised learning was a zero signal spiking to one at fR peaks marked by the provided correct annotations. The training is performed in a single batch and produces $\mathbf{W}^\mathrm{out}$ that are optimal in the sense of quadratic error between $y(t)$ and $y^\mathrm{target}(t)$. After training $\mathbf{W}^\mathrm{out}$ remain fixed.

The ESN can be seen as a big (N_x being in order of hundreds or thousands) nonlinear expansion with memory $\mathbf{x}(n)$ (1)(2) of the input signals $\mathbf{u}(t)$ followed by an optimal linear combination (3) to produce the output signal $y(t)$ as close to the target $y^\mathrm{target}(t)$ as possible.

3.2. Probabilistic interpretation of the indicator signals

The final component in our architecture is responsible for interpreting the fR indicator signal $y(t)$ to produce discrete fR annotations $\{t_1, t_2, \ldots, t_m\}$ where m is not fixed.

Because the data is noisy and varied, the indicator signals $y(t)$ produced by the ESN after training are not perfect and quite differ from the corresponding desired clean $y^\mathrm{target}(t)$. Still, on average $y(t)$ is hopefully higher at time points t corresponding to fR peaks.

To deal with this imprecision and be able to combine it with other sources of information using a probabilistic framework, we interpret the signal $y(t)$ as an indication of the probability $P(t|\mathbf{u})$ that there is an fR peak event at a time step t, given data $\mathbf{u}$. We must keep in mind, however, that $y(t)$ is not exactly trained to be the probability and especially the scaling of $y(t)$ can be way off, thus it should only be used as a comparative, but not an absolute value. We thus denote $P(t|\mathbf{u}) = f(y(t))$, where $f(\cdot)$ is a monotonic function, details of which we will discuss in Section 3.3. We trimmed negative values of $y(t)$ as a practical means to ensure non-negativity. Note also, that a probability of an event at a certain time step t is related to a probability density function over time by a constant dt which in our case is 1 ms.

In this probabilistic framework we want to find a sequence of fR annotations $\{t_1, t_2, \ldots, t_m\}$ that maximizes the probability of fR peak events occurring at those time instances, given the (evidence) data $\mathbf{u}$:

$$\{t_1, t_2, \ldots, t_m\} = \underset{\{t'_1, t'_2, \ldots, t'_m\}}{\arg\max} P(t'_1, t'_2, \ldots, t'_m|\mathbf{u})^{\frac{1}{m}}, \tag{4}$$

where $\{t'_\cdot\}$ are candidate annotations bound by $\arg\max$. Since the number of events m is not fixed here and the probabilities involved are typically < 1, the normalization by $1/m$ is introduced in (4) to remove a bias toward smaller m.

In the following subsections we will present increasingly advanced algorithms for combining the available information in producing the fR annotations.

3.2.1. Direct interpretation. The most straightforward approach of annotation is setting a threshold y_{th} and annotating every t_i with an fR event where $y(t_i) > y_{\text{th}}$. This would work but only for very clean signals, and setting of y_{th} would be problematic.

A probabilistic interpretation of this approach would be based on an assumption that fR events at time instances t_i are statistically independent between each other, and thus

$$P(t_1, t_2, \ldots, t_m|\mathbf{u}) = \prod_{i=1}^{m} P(t_i|\mathbf{u}) = \prod_{i=1}^{m} f(y(t_i)). \tag{5}$$

Such approach would pick annotations $\{t_i\}$ with highest $y(t_i)$ ($> y_{\text{th}}$) that maximizes $\prod_{i=1}^{m} y(t_i)$.

However, as in speech recognition, we can employ our knowledge of higher level statistics to improve the interpretation of information coming from lower level noisy sound signal by providing expectations and context to it. In speech recognition these higher level statistics would be context-specific dictionaries, probabilistic grammars, etc. In our task this would be the available statistics of fR events. The idea is sketched in Figure 2 right.

3.2.2. Employing R-R interval statistics. The direct method treats fR events at t_{i-1} and t_i as statistically independent. However, we know that fetal R-R (fRR) interval durations ($t_i - t_{i-1}$) typically lie in certain ranges. A more intelligent and widely popular approach is to set a permissible interval τ_{RR} for the R-R values and find the next annotation at maximum inside the permissible interval:

$$t_i = \underset{t'_i \in (t_{i-1}+\tau_{\text{RR}})}{\arg\max} P(t'_i|\mathbf{u}) = \underset{t'_i \in (t_{i-1}+\tau_{\text{RR}})}{\arg\max} y(t'_i). \tag{6}$$

This approach aims at maximizing (5) with this additional hard constraint between every subsequent t_{i-1} and t_i.

This method worked better. This is in fact how our mQRS detection mentioned in Section 2.3 was implemented, except detecting maximum on one of the leads of the the raw signal $\mathbf{u}(n)$ instead of the trained indicator $y(t)$.

We can, however, make better use of the fRR statistics. We compute a histogram of fRR durations from the Dataset A annotations and, after removing some outliers, we model their distribution as a Gaussian $\mathcal{N}\left(\mu_{f\text{RR}}, \sigma^2_{f\text{RR}}\right)$ with estimated parameters $\mu_{f\text{RR}} = 424\,\text{ms}$ and $\sigma_{f\text{RR}} = 40\,\text{ms}$ (Figure 2 top left). We thus estimate the probability of fRR duration being $(t_i - t_{i-1})$, instead of setting a hard interval, which is also the probability that fR occurs at t_i, given that it occurred at t_{i-1}: $P(f\text{RR} = t_i - t_{i-1}) = P_{f\text{RR}}(t_i - t_{i-1}) = P(t_i|t_{i-1})$.

Under this assumption that probability of every t_i is dependent not only on the

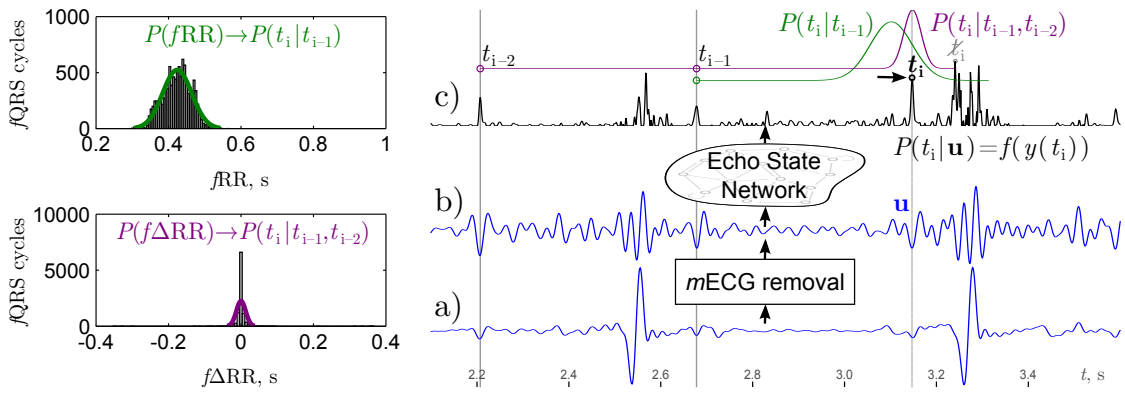

Figure 2. Left: fRR and $f\Delta$RR histograms and fitted Gaussian distribution models. Right: a sketch on how fRR event statistics are employed to refine fR event recognition. a) The filtered raw signal. b) The signal with mECG (imperfectly) removed. c) The indicator signal and its interpretation employing fRR statistics. Only one of four leads is shown in a) and b) for visual clarity.

input $\mathbf{u}$, but also on t_{i-1},

$$P(t_1, t_2, \ldots, t_m|\mathbf{u}) = \prod_{i=1}^{m} P(t_i|t_{i-1}, \mathbf{u}). \tag{7}$$

Assuming that t_{i-1} is already decided for and no longer depends on $\mathbf{u}$ (a "Markov blanket"), and applying Bayes' rule, we get

$$P(t_i|t_{i-1}, \mathbf{u}) = \frac{1}{\bar{P}_t} P(t_i|t_{i-1}) P(t_i|\mathbf{u}), \tag{8}$$

where $P(t_i) = \bar{P}_t$ is a constant average probability of fR occurring at any t_i.

Putting (4), (7), and (8) together, in this case we have to maximize

$$\left(\prod_{i=1}^{m} P\left(t_i|t_{i-1}\right) P(t_i|\mathbf{u}) \right)^{\frac{1}{m}} = \left(\prod_{i=1}^{m} P\left(t_i|t_{i-1}\right) f(y(t_i)) \right)^{\frac{1}{m}} \tag{9}$$

A simple dynamic algorithm to do this is to select next t_i as

$$t_i = \underset{t'_i \in (t_{i-1}+\tau_{\mathrm{RR}})}{\arg\max} \left(P(t'_i|t_{i-1}) y(t'_i) \right) \tag{10}$$

with the interval τ_{RR} corresponding to the range where $P(t'_i|t_{i-1})$ is high enough to consider. The algorithm is really fast, it has time complexity $\mathcal{O}(T)$, just like (6), where T is the length of the signal. It is, however, imprecise, since it greedily chooses annotations with high probability even if they later lead to annotations with low probability.

To enable us working with potentially infinite sequences, and to simplify computations, let us switch from the global geometric average presented in (4) to a local in time exponential geometric moving average, and redefine (9) in an iterative form

$$P\left(\ldots, t_{i-1}, t_i|\mathbf{u}\right) = P\left(\ldots, t_{i-2}, t_{i-1}|\mathbf{u}\right)^{1-\gamma} \left(P\left(t_i|t_{i-1}\right) P(t_i|\mathbf{u}) \right)^{\gamma}, \tag{11}$$

where $\gamma < 1$ is an exponential decay factor.

Let $p_{\max}(t)$ be the maximal probability of a sequence of annotations leading up to, and including t as the last annotation in the sequence:

$$p_{\max}(t) = \max_{\{\ldots, t'_{i-2}, t'_{i-1}\}} P\left(\ldots, t'_{i-2}, t'_{i-1}, t|\mathbf{u}\right). \tag{12}$$

$p_{\max}(t)$ can also be computed iteratively:

$$p_{\max}(t) = \max_{t'_{i-1} \in (t-\tau_{\mathrm{RR}})} \left(p_{\max}(t'_{i-1})^{1-\gamma} P\left(t|t'_{i-1}\right)^{\gamma}\right) P(t|\mathbf{u})^{\gamma}. \tag{13}$$

This way for a finite signal we can obtain $p_{\max}(t)$ for every t, going forward in time, find a good candidate for t_m, giving a high $p_{\max}(t_m)$ at the end of the signal, and retrace the sequence $t_m, t_{m-1}, \ldots, t_1$ that produced it, going backward in time – thus obtaining the annotation sequence that maximizes (11).

Note that we only need to consider an interval $(t - \tau_{\mathrm{RR}})$ for t'_{i-1} in (13) where the Gaussian $P(t|t'_{i-1})$ is sufficiently large, i.e., a reasonable interval for an fRR durations. Time complexity of this algorithm is $\mathcal{O}(T \cdot |\tau_{\mathrm{RR}}|)$, where T is the length of the signal and $|\tau_{\mathrm{RR}}|$ is the range of possible fRR durations.

3.2.3. Employing R-R interval variation statistics. The methods above treated durations of neighboring fRR intervals as statistically independent. However, we know that this is not the case. We compute a histogram of changes $f\Delta$RR between durations of all two subsequent fRR intervals, and see that (excluding several outliers) it roughly follows a Gaussian distribution $\mathcal{N}\left(0, \sigma^2_{f\Delta\mathrm{RR}}\right)$ with estimated $\sigma_{f\Delta\mathrm{RR}} = 13\,\mathrm{ms}$ (Figure 2 bottom left). This is a much narrower distribution than that of fRR. Thus modeled probability that a change in fRR duration will be $\big((t_i - t_{i-1}) - (t_{i-1} - t_{i-2})\big) = (t_i - 2t_{i-1} + t_{i-2})$ is taken as the probability of fR at t_i, given fR at t_{i-1} and t_{i-2}: $P(f\Delta\mathrm{RR} = t_i - 2t_{i-1} + t_{i-2}) = P(t_i|t_{i-1}, t_{i-2})$.

Under this assumption that probability of every t_i is dependent not only on the input $\mathbf{u}$, but also on t_{i-1} and t_{i-2},

$$P(t_1, t_2, \ldots, t_m|\mathbf{u}) = \prod_{i=1}^{m} P(t_i|t_{i-1}, t_{i-2}, \mathbf{u}). \tag{14}$$

Assuming that t_{i-1} and t_{i-2} are already decided for and no longer depend on $\mathbf{u}$ (a "Markov blanket"), and applying Bayes' rule and the result from (8), we get

$$P(t_i|t_{i-1}, t_{i-2}, \mathbf{u}) = \frac{1}{\overline{P}_t} P(t_i|t_{i-1}, t_{i-2}) P(t_i|\mathbf{u}), \tag{15}$$

where $P(t_i)$ is a constant average probability of fR occurring at any t_i.

Getting back to the geometric moving average and an iterative form,

$$P\left(\ldots, t_{i-1}, t_i|\mathbf{u}\right) = P\left(\ldots, t_{i-2}, t_{i-1}|\mathbf{u}\right)^{1-\gamma} \left(P\left(t_i|t_{i-1}, t_{i-2}\right) P(t_i|\mathbf{u})\right)^{\gamma}, \tag{16}$$

where $\gamma < 1$ is an exponential decay factor.

Let $p_{\max}(t, t_-)$ be the maximal probability of a sequence of annotations leading up to, and including t_- and t as the last two annotations in the sequence

$$p_{\max}(t, t_-) = \max_{\{\ldots, t'_{i-3}, t'_{i-2}\}} P\left(\ldots, t'_{i-3}, t'_{i-2}, t_-, t|\mathbf{u}\right). \tag{17}$$

$p_{\max}(t, t_-)$ can also be computed iteratively:

$$p_{\max}(t, t_-) = \max_{t'_{i-2} \in (2t_- - t - \tau_{\Delta\mathrm{RR}})} \left(p_{\max}(t_-, t'_{i-2})^{1-\gamma} P\left(t|t_-, t'_{i-2}\right)^{\gamma}\right) P(t|\mathbf{u})^{\gamma}. \tag{18}$$

This way, for a finite signal we compute all $p_{\max}(t, t_-)$ in the forward direction for every $t \in (0, T]$ and $t_- \in (t - \tau_{\mathrm{RR}})$, saving $\arg\max t'_{i-2}$ from (18) for every pair (t, t_-). We take $t_m = \arg\max_t \max_{t_-} p_{\max}(t, t_-)$ at the end of the signal and trace back the sequence $t_m, t_{m-1}, \ldots, t_1$ that produced it, to get the fR annotations. The interval $\tau_{\Delta\mathrm{RR}}$ in (18) is centered around 0 where $P(t|t_{-1}, t_{-2})$ has high enough values.

Time complexity of this algorithm is $\mathcal{O}(T \cdot |\tau_{\mathrm{RR}}| \cdot |\tau_{\Delta\mathrm{RR}}|)$, where T is the length of the signal and $|\tau_{\mathrm{RR}}|$ is the range of possible fRR durations and $|\tau_{\Delta\mathrm{RR}}|$ is the range of possible $f\Delta$RR variations. Space complexity is $\mathcal{O}(T \cdot |\tau_{\mathrm{RR}}|)$ to store $p_{\max}(t, t_-)$ and t'_{i-2}'s for back-tracking.

3.2.4. Employing both statistics. By the way we have defined $P(t_i|t_{i-1}, t_{i-2})$ in Section 3.2.3, we assume that fRR durations change following a random walk process of Brownian motion. In this definition there is no restriction to possible fRR durations or bias toward more probable fRR values, like $P(t_i|t_{i-1})$ of Section 3.2.2. In practice fRR durations are restricted by τ_{RR} in (18), but can still take improbable values within it. To introduce the bias $P(t_i|t_{i-1})$ back, we can combine the two algorithms (13) and (18) into

$$p_{\max}(t, t_-) = \max_{t'_{i-2} \in (2t_- - t - \tau_{\Delta\mathrm{RR}})} \left(p_{\max}(t_-, t'_{i-2})^{1-\gamma} P\left(t|t_-, t'_{i-2}\right)^{\lambda\gamma}\right) P(t|\mathbf{u})^{\gamma} P(t|t_-)^{\kappa\gamma} \tag{19}$$

with appropriate powers λ and κ that can be tuned empirically to balance the two biases. The algorithm essentially retains the same properties as (18) and falls back to it in case $\kappa = 0$.

An even more principled approach would be to model $P(t_{i-1} - t_{i-2}, t_i - t_{i-1})$ as a joint two-dimensional Gaussian distribution and compute $P(t_i|t_{i-1}, t_{i-2})$ as a conditional probability from it, which we leave as a future work.

3.3. Parameter tuning and validation

We used 15-fold cross-validation on the 75 Dataset A annotated signals with an error function that measures mean square distance between individual fR annotations similar to Events 2 and 5 of the Challenge to test the many design options and parameters in

all the components of our solution. ESNs enable us to do this massive cross-validation with minimal overhead (Lukoševičius 2012). Our solution was implemented in Matlab.

Parameters for data prerocessing and mECG removal are discussed in Section 2.

We have tried different parameter settings with our ESN networks (2)(3) following practices described in (Lukoševičius 2012). We used ESN reservoirs of size $N_{\mathrm{x}} = 1000$ or 500, leaking rate $\alpha = 0.9$, spectral radius of the reservoir connections $\rho(\mathbf{W}) = 0.9$ or 0.94, $\mathbf{W}^{\mathrm{in}}$ scaling of 0.1 or 0.08. In some of our solutions we used several (up to five) ESNs by training and running them in parallel, then averaging their outputs $y(t)$.

We used the indicator signals $y(t)$ produced by the cross-validation (by the ESNs that are not directly trained on these signals) to tune our probabilistic algorithms, the latter are however not truly cross-validated, since the parameters are set directly by hand and are not meta-parameters for learning.

We refined our theoretically derived algorithms of Section 3.2 with power coefficients similar to the ones in (19) that allowed for balancing the influence of the three sources of probabilities: $P(t|\mathbf{u}) = f(y(t))$, $P(t_i|t_{i-1})$, and $P(t_i|t_{i-1}, t_{i-2})$. The parameters were coarsely hand-tuned to empirically improve performance. For the Gaussian distributions they are equivalent to changing their standard deviations away from the estimated ones.

Two parameters like in (19) are enough to balance three probabilities, because power is a monotonous operation and thus maximum is invariant to it. This way the power coefficient next to $P(t|\mathbf{u})$ would be redundant: its relative strength depends on the other two, and the absolute power scaling does not matter. Also, notice that a correct absolute scaling of the probabilities involved in all our algorithms is not necessary. We take $P(t|\mathbf{u}) = f(y(t)) = a\, y(t)^b$, where a and b are some scalar coefficients, and can reduce it to $P(t|\mathbf{u}) = y(t)$ without loss of generality, because a does not matter and b is empirically optimized for when tuning κ and λ in (19).

4. Results

We have tested many modifications of our solution, especially different options and parameters of the probabilistic DP algorithms described in Section 3.2. Since the number of solutions that Challenge organizers would accept to test on the hidden testing data was very limited, we did most of the scoring internally on the available training Dataset A, using cross-validation when possible as described in Section 3.3.

Internal (A75) and official scores (where available) for our different probabilistic DP algorithms are presented in Table 1. As mentioned, A75 scores are similar to and correlate with Event 5, but has a much bigger scaling as it sums squared distances in milliseconds.

The best official Challenge scores are with a mention of the place they have ranked in the Challenge. Event 4 is a mean squared error (MSE) scored in the domain of fetal heart rate (fHR) obtained from the annotations, and Event 5 is root MSE in raw fRR interval durations between the produced and the correct annotations of Dataset B (Silva

Table 1. Internal (A75) and challenge error scores with different algorithms.

Algorithm	A75	Event 4	Event 5
fRR flat greedy (6)	523338		
fRR Gauss greedy (10)	618214	66.327 (7^{th})	11.027
fRR flat DP (13)	400900		
fRR Gauss DP (13)	397777		
$f\Delta$RR Gauss DP (18)	183113	254.143	8.675
$f\Delta$RR +fRR Gauss DP (19)	178585	147.236	8.239 (5^{th})

et al. 2013), similar to our internal A75.

The algorithm "fRR flat DP" in Table 1 is a version of dynamic programming algorithm (13), where $P(t_i|t_{i-1})$ is uniform (flat) within an admissible interval τ_{RR}.

All the A75 scores in Table 1 are produced from the same $y(t)$ signals, coming from an ESN of size $N_{\text{x}} = 500$, for a better comparison (but not top performance). The results of the Challenge Events 4 and 5, on the other hand, do not reflect the difference of performance of the different algorithms precisely, because striving for the best performance and having limited tries to obtain official scores, some of the parameters and details vary among the submissions. They are thus also not directly comparable to A75 of the corresponding algorithms, even though A75 are produced with the algorithms that were made to resemble the submitted ones.

An observation can be made that algorithms taking into account fRR statistics are less likely to wonder into improbable fHR values and thus have better Event 4 scores, while those using $f\Delta$RR statistics tend to track every fR more precisely (when at all) and thus perform better in the fRR domain: A75 and Event 5.

The approach A75 which we chose to internally measure performance highly favors the latter. We can observe in Table 1 that using Gaussian $P(t_i|t_{i-1})$ distributions instead of flat ones within permissible values, does little to improve the A75 score, and sometimes is even detrimental, except in the best final solution. Indeed, when tuning power coefficient κ next to $P(t_i|t_{i-1})$, the best A75 scores where often obtained close to when $\kappa \to 0$ and thus the distribution approaches uniform. This may be related to the fact that fRR durations in the signals are highly correlated and multiplying the $P(t_i|t_{i-1})$ many times assigns a too low probability to an atypical fHR.

On the other hand, Event 4 scores are highly boosted by taking the $P(t_i|t_{i-1})$ bias into consideration. It is revealed that due to specifics of the scoring system of the Challenge, a trivial solution with uniformly constantly placed fR annotations producing a mean fHR scores quite high (Behar, Oster & Clifford 2013). Thus $P(t_i|t_{i-1})$ bias might be even more important for difficult unseen data to score high. For practical purposes, however, such high scores could be deceptive and the use of the fRR and $f\Delta$RR statistics should balance the correctness of annotations with their temporal smoothness.

Table 2 represents some additional results, changing other parts of our architecture with the best performing algorithm from Table 1. This gives a better understanding on

what factors contribute to the good performance most.

Table 2. Effects of different types of preprocessing of data.

Algorithm	A75
$N_x = 100$ ESN, mECG cancellation	237953
$N_x = 500$ ESN, mECG cancellation	178585
$N_x = 1000$ ESN, mECG cancellation	184990
$N_x = 500$ ESN, no mECG cancellation	709196
$N_x = 1000$ ESN, no mECG cancellation	636630

We can see that using a significantly smaller ESN reservoir is less detrimental to the performance than using a less powerful DP algorithm (Table 1). It is unusual in this case that performance does not improve than making the reservoir bigger ($N_x = 1000$), which should most probably be attributed to the fact that the parameters we (over-)tuned specifically for $N_x = 500$. We have also experimented with removing the mECG cancellation part from our architecture entirely and report results in Table 2.

5. Discussion and conclusions

Our approach scored high among 53 other contestants in the Challenge. It is also quite fast: our slowest submission (19) took about 30 s to process one signal on an up-to-date personal computer, i.e., twice the real time. The algorithms could easily be made more precise sacrificing the speed, by making ESNs bigger and/or using more of them.

Using the statistics of the R events biases the system to a usual scenario: it improves correctness of annotations in healthy settings, but might be less suited to recognize abnormal events, which might be more important. This should be kept in mind, and emergency events prioritized if necessary by adapting the statistical models. Our statistical framework allows to do this explicitly by modifying $P(t_i|t_{i-1})$ and $P(t_i|t_{i-1}, t_{i-2})$.

Our approach also did not produce the shape of fECG, but it could be done from the fR annotations and the preprocessed signal in the same way as for mECG. Alternatively, ESN could probably learn to extract fECG directly, given the reference training signals.

Another improvement would be to use logistic regression and readout instead of the linear one (3), which would be better suited for the probabilistic interpretation. For now we have chosen the former because it has a closed-form solution and is computationally much cheaper.

Since the advanced versions of our DP algorithms uses back-tracking, their adaptation to continuous real-time monitoring should analyze the data in time windows or refine the previous annotations by backtracking from time to time.

We see in Table 2 that the results without mECG cancellation described in Section 2.3 are comparable with weaker $y(t)$ interpretation algorithms from Table 1. This

demonstrates robustness of our approach and confirms the notion, that in an intelligent system, stronger incorporation of high level knowledge can help to deal with even extreme levels of noise in low level sensory information.

Our fQRS detection is also quite generic and could be used for other types of signals and annotations, without much hand-crafting because it uses machine learning and statistics estimated from data.

Open source code of our approach compatible with the Challenge scoring scripts is made available under `http://minds.jacobs-university.de/mantas/fetalQRS`.

Acknowledgements

This research is supported by European Social Fund within the project "Intelligent wearable sensor system for human health monitoring" (Agreement No. VP1-3.1-SMM-10-V-02-004).

References

Andreotti, F., M. Riedl, T. Himmelsbach, D. Wedekind, S. Zaunseder, N. Wessel & H. Malberg. 2013. Maternal signal estimation by Kalman filtering and Template Adaptation for fetal heart rate extraction. In *Computing in Cardiology Conference (CinC), 2013*. pp. 193–196.

Behar, J., J. Oster, Qiao Li & G.D. Clifford. 2013. "ECG Signal Quality During Arrhythmia and Its Application to False Alarm Reduction." *Biomedical Engineering, IEEE Transactions on* 60(6):1660–1666.

Behar, Joachim, Alistair Johnson, Gari D. Clifford & Julien Oster. 2014. "A Comparison of Single Channel Fetal ECG Extraction Methods." *Annals of Biomedical Engineering* 42:1340–1353.

Behar, Joachim, Julien Oster & Gari D. Clifford. 2013. Non-Invasive FECG Extraction from a Set of Abdominal Sensors. In *Computing in Cardiology Conference (CinC), 2013*. pp. 297–300.

Camps-Valls, Gustavo, Marcelino Martinez-Sober, Emilio Soria-Olivas, Rafael Magdalena-Benedito, Javier Calpe-Maravilla & Juan Guerrero-Martinez. 2004. "Foetal ECG recovery using dynamic neural networks." *Artificial Intelligence in Medicine* 31(3):197 – 209.
URL: `http://www.sciencedirect.com/science/article/pii/S0933365704000508`

Clifford, Gari D., Ikaro Silva, Joachim Behar & George Moody. 2014. "Editorial: Noninvasive Fetal ECG analysis." *Physiological Measurement* p. in press.

Farvet, A.G. 1968. "Computer Matched Filter Location of Fetal R-Waves." *Medical & Biological Engineering* 6 (no. 5):467–475.

Jaeger, Herbert. 2001. The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148 German National Research Center for Information Technology.

Jaeger, Herbert. 2007. "Echo state network." *Scholarpedia* 2(9):2330.
URL: `http://www.scholarpedia.org/article/Echo_state_network`

Kanjilal, P.P., S. Palit & G. Saha. 1997. "Fetal ECG extraction from single-channel maternal ECG using singular value decomposition." *Biomedical Engineering, IEEE Transactions on* 44(1):51–59.

Lathauwer, Lieven De, Bart De Moor, Joos Vandewalle, Girona Spain, L. De Lathauwer, D. Callaerts & B. De Moor. 1994. Fetal Electrocardiogram Extraction by Source Subspace Separation. In *In Proc. HOS'95*. pp. 134–138.

Lukoševičius, M. & V. Marozas. 2013. Noninvasive fetal QRS detection using Echo State Network. In *Computing in Cardiology Conference (CinC), 2013*. pp. 205–208.

Lukoševičius, Mantas. 2012. A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade, 2nd Edition*, ed. Grégoire Montavon, Geneviève B. Orr & Klaus-Robert Müller. Vol. 7700 of *LNCS* Springer pp. 659–686.

Lukoševičius, Mantas & Herbert Jaeger. 2009. "Reservoir computing approaches to recurrent neural network training." *Computer Science Review* 3(3):127–149.

Martens, Suzanna M M, Chiara Rabotti, Massimo Mischi & Rob J Sluijter. 2007. "A robust fetal ECG detection method for abdominal recordings." *Physiological Measurement* 28(4):373.

Petrėnas, A., V. Marozas, L. S ornmo & A. Lukoševičius. 2012. "An Echo State Neural Network for QRST Cancellation During Atrial Fibrillation." *Biomedical Engineering, IEEE Transactions on* 59(10):2950–2957.

Sameni, Reza & Gari D. Clifford. 2010. "A Review of Fetal ECG Signal Processing; Issues and Promising Directions." *The Open Pacing, Electrophysiology & Therapy Journal (TOPETJ)* 3:4–20.

Silva, I., J. Behar, R. Sameni, Tingting Zhu, J. Oster, G.D. Clifford & G.B. Moody. 2013. Noninvasive Fetal ECG: the PhysioNet/Computing in Cardiology Challenge 2013. In *Computing in Cardiology Conference (CinC), 2013*. pp. 149–152.

Widrow, B., Jr. Glover, J.R., J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, Jr. Eugene Dong & R.C. Goodlin. 1975. "Adaptive noise cancelling: Principles and applications." *Proceedings of the IEEE* 63(12):1692–1716.