UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale in Ingegneria Biomedica

Sviluppo di un algoritmo per il processamento di dati provenienti da emogasanalisi: caso di studio terapia intensiva neonatale

Development of an algorithm for the processing of data from blood gas analysis: neonatal intensive care case report

Relatore: Chiar.mo

Prof. Lorenzo Scalise

Tesi di Laurea di:

Lorenzo Paoletti

Correlatori:

Chiar.mo Prof. Virgilio Paolo Carnielli
Ing. Paolo Marchionni

A.A. 2019 /2020

Vorrei ringraziare il Professor Lorenzo Scalise ed il Professor Enrico Primo Tomasini, per la loro disponibilità e per avermi dato la possibilità di intraprendere questo tirocinio che mi ha portato alla realizzazione di questa tesi di laurea.

Ringrazio l'ingegner Paolo Marchionni e Luca Antognoli per avermi seguito passo passo con la più assoluta disponibilità, e per tutte le dritte e i consigli e l'assistenza che mi hanno dato.

Ringrazio il Professor Virgilio Carnielli per la disponibilità e l'accesso che mi ha fornito al suo reparto di Terapia Intensiva Neonatale.

Un grazie a tutto il personale della TIN del Salesi per la collaborazione.

Grazie a tutti amici che mi hanno fatto svagare e divertire quando più ne avevo bisogno.

E soprattutto il GRAZIE più importante va alla mia famiglia che mi ha sostenuto e su(o)pportato per tutto questo tempo.

Indice

1 INTRODUZIONE 1

2 LA STRUMENTAZIONE E LA TIN 2

2.1 METODI DI MISURA 3
2.2 I PARAMETRI 5

3 PROCESSAMENTO DEI DATI 8

3.1 I DATI 8
3.2 PULIZIA DEI DATI 10
3.3 FUORI RANGE ED ERRORI DI MISURA 14
3.3.1 Indagine su possibili correlazioni 14
3.3.2 Statistiche 16
3.4 T-TEST 18
3.5 CLUSTERING TRAMITE K-NEAREST NEIGHBORS 21

4 CONCLUSIONI E SVILUPPI FUTURI 25

5 APPENDICE CODICI MATLAB 26

6 BIBLIOGRAFIA E SITOGRAFIA 38

1 INTRODUZIONE

Questa tesi è figlia del tirocinio nato grazie alla collaborazione tra il Dipartimento di Ingegneria Industriale e Scienze Matematiche (DIISM) della facoltà di Ingegneria ed il reparto di Terapia Intensiva Neonatale (TIN) dell'ospedale pediatrico Salesi di Ancona.

In particolare, il lavoro qui svolto riguarderà i dati provenienti dalle emogasanalisi (EGA) effettuate sui neonati pretermine stazionati nel reparto di Terapia Intensiva Neonatale.

I problemi principali che hanno fatto scaturire la necessità di questo studio sono stati la difficoltà nella gestione e consultazione dei dati causata dalla formattazione in cui si presentano in uscita dall'apparecchio e la sospetta presenza di artefatti da prelievo, ovvero la sospetta presenza di disturbi dovuti alla strumentazione o dovuti ad errore umano che influiscono in maniera negativa sul valore dei parametri misurati nell' EGA e che di conseguenza rendono non affidabile l'esame in questione. Tali interferenze alla misura possono essere causate da fattori esterni come la presenza di precipitati o sostanze estranee, da malfunzionamenti della strumentazione, da errori umani e dalla difficoltà del prelievo in sé, che ricordo in questo caso essere effettuato su neonati pretermine stazionati in terapia intensiva.

L'obbiettivo di questa tesi è quindi quello di creare uno script in ambiente MATLAB che permetta di gestire ed analizzare efficacemente una grande mole di dati in formato *.xls*, di attuare operazioni di pulizia dei dati e che permetta di discriminare le EGA in base a parametri arbitrari secondo le necessità del personale medico, con particolar interesse nei dati soggetti ad artefatti da prelievo.

Il primo step sarà quello di comprendere i dati in esame, la loro tipologia e le problematiche a loro associate in modo da capire le direzioni da prendere nello sviluppo dell'algoritmo in ambiente Matlab. In secondo luogo, avverrà l'elaborazione vera e propria dei dati e la loro trasposizione in un formato leggibile e lavorabile in Matlab. Una volta che i dati saranno nel formato adatto, si andrà alla ricerca di un metodo affidabile per la discriminazione delle EGA.

2 LA STRUMENTAZIONE E LA TIN

La Terapia Intensiva Neonatale è il reparto in cui vengono stazionati ed assistiti i neonati di età inferiore a trenta giorni che necessitano di assistenza particolarmente complessa ed avanzata a causa di nascite pretermine, difetti congeniti e/o situazioni a rischio. I bambini qui stazionati vengono quindi sottoposti ad un costante monitoraggio multiparametrico in modo da poter intervenire rapidamente in caso di un eventuale peggioramento delle condizioni.

L'emogasanalisi in particolare è un test usato principalmente per valutare l'efficienza respiratoria del soggetto in esame e anche per controllarne l'equilibrio acido-base. Si basa principalmente sull'analisi di un prelievo ematico di tre tipi: venoso, arterioso e capillare. Il prelievo avviene su apposite siringhe o provette capillari rivestite di sostanza anticoagulante, trattandosi di neonati il prelievo viene ottenuto anche dal cordone ombelicale.

L'apparecchiatura per l'emogasanalisi utilizzata nella TIN al momento del mio studio è un analizzatore della serie ABL dell'azienda danese Radiometer. L'apparecchio della serie ABL è un analizzatore che permette di misurare parametri come il pH, le pressioni parziali dei gas e le concentrazioni delle sostanze presenti nel sangue; parametri che permettono di valutare l'efficienza respiratoria del soggetto in esame. Nel paragrafo successivo illustrerò il suo funzionamento base ed elencherò i parametri principali.

2.1 METODI DI MISURA

La strumentazione in questione utilizza due metodi di misura: tramite elettrodi e tramite un sistema ottico.

Uno schema degli elettrodi utilizzati si può vedere in (Fig 1) dove si distinguono:

- Un contatto elettrico
- Una colorazione per semplificarne il riconoscimento
- Un rivestimento per supporto e protezione
- Una soluzione elettrolitica che provvede al contatto elettrico tra l'elettrodo ed il campione
- Una membrana per separare il campione dall'elettrodo, sensibile alle varie sostanze in esame

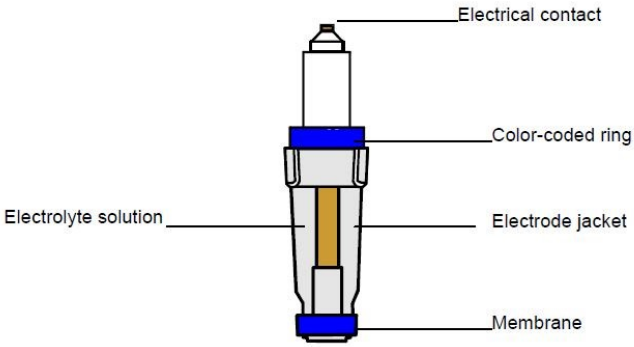


Fig. 1-Schema dell'elettrodo

I due principi di misura utilizzati con gli elettrodi sono la potenziometria e l'amperometria.

Il metodo potenziometrico si basa sulla misura del potenziale del sistema mostrato in figura 2 qui di seguito

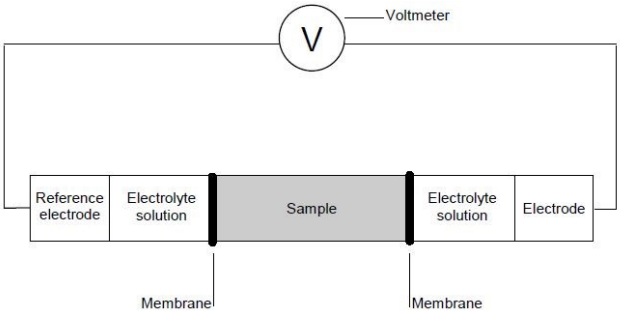


Fig. 2-Schema del metodo potenziometrico

L'elettrodo di riferimento (Reference electrode) è composto da una barra in argento con soluzione elettrolitica di formiato di sodio (HCOONa) stabilizzato a pH 5,5 tramite acido idrocloridrico.

Utilizzando l'equazione di Nernst il potenziale misurato è correlato all'attività chimica del campione che a sua volta viene convertita nella concentrazione.

$$E_{sample} = E_0 + \frac{2.3RT}{nF} \log a_x \qquad a_x = \gamma\, c_x$$

Il metodo amperometrico (Fig. 3) sfrutta la riduzione e l'ossidazione su anodo e catodo: Applicata una tensione costante, la specie chimica in esame che si riduce sul catodo genera una corrente elettrica misurata da un amperometro. Tale corrente è proporzionale alla concentrazione cercata.

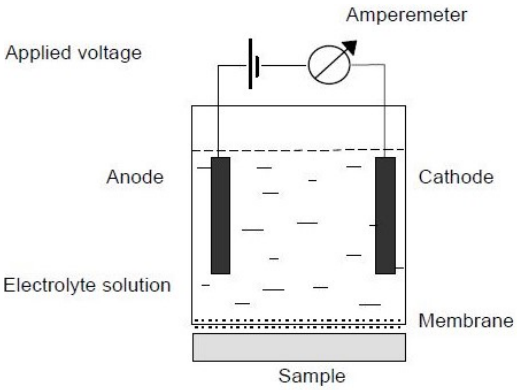


Fig. 3-schema del metodo amperometrico

L'altro sistema, il sistema ottico, si basa su uno spettrofotometro a 128 lunghezze d'onda con un range di misura di 478-672nm collegato tramite fibra ottica ad un emolizzatore e ad una camera di misura. (Fig.4)

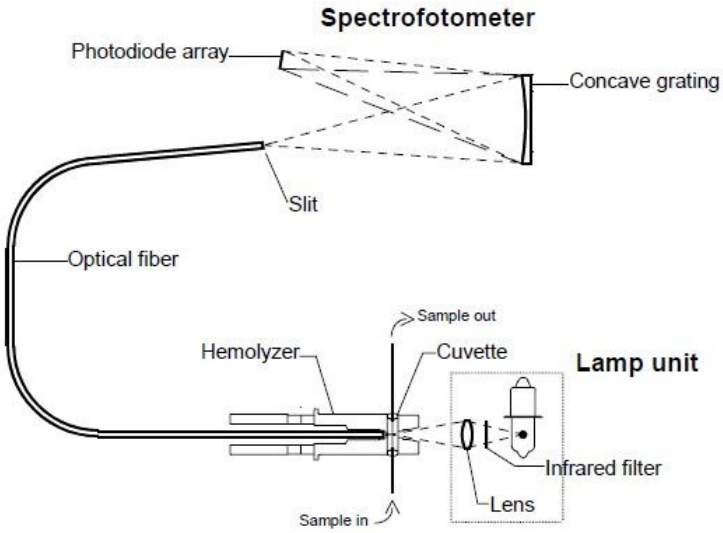


Fig. 4-Schema del sistema ottico

Il metodo usato dal sistema ottico in questione è la spettroscopia di assorbimento del visibile: 1µL del campione di sangue, una volta trasportato nella cuvetta regolata a 37˚, viene emolizzato tramite ultrasuoni ad una frequenza di circa 30kHz e a una sovrapressione di 1atm. Il sangue emolizzato viene attraversato da luce di intensità costante filtrata nell'infrarosso, luce che viene poi divisa in 128 lunghezze d'onda e riflessa sull'array di 128 fotodiodi, uno per ogni lunghezza d'onda. Ogni diodo converte l'intensità di luce ricevuta in segnale elettrico. I segnali elettrici e quindi le intensità luminose vengono misurate in ogni diodo formando le basi per lo spettro di assorbimento, analizzato dal computer integrato che restituisce quindi i parametri ossimetrici.

Legge di lambert-Beer: L'assorbimento luminoso di un composto è direttamente proporzionale alla concentrazione del composto e alla lunghezza del campione che la luce attraversa.

$$A_y^\lambda = \varepsilon_y^\lambda \times c_y \times l$$

Nella formula *A* è l'assorbimento, ε è un coefficiente, *c* la concentrazione, l la lunghezza del campione, λ la lunghezza d'onda e y il campione. L'assorbimento è definito a sua volta dalla formula:

$$A = \log \frac{I_0}{I}$$

Dove I_0 è l'intensità della luce trasmessa attraverso acqua e I l'intensità della luce trasmessa attraverso il campione.

Altre sostanze presenti nel sangue che assorbono la luce nelle stesse lunghezze d'onda dei parametri di interesse andranno ad interferire sullo spettro di assorbimento se non tenute in considerazione. L'analizzatore compensa comunque in automatico le interferenze degli: Intralipidi, solfoemoglobina e SHb.

2.2 I PARAMETRI

Avendo visto nel paragrafo precedente in che modo l'apparecchio misura i vari parametri di interesse, esporrò ora quali sono questi parametri, nello specifico solo quelli che saranno presenti nel lavoro qui svolto in modo tale da avere un elenco di riferimento dei parametri e loro relative abbreviazioni di cui rispettivamente parlerò e farò uso in seguito.

L'analizzatore restituisce più di settanta parametri categorizzabili principalmente in: parametri di ossigeno, ossimetrici, acido-base, bilirubina, elettroliti e metaboliti. Non tutti i parametri vengono direttamente misurati tramite i metodi illustrati nel paragrafo precedente, alcuni vengono infatti calcolati in base al valore di altri cioè vengono calcolati dalla strumentazione con equazioni e formule che utilizzano valori di altri parametri, siano essi misurati o a loro volta calcolati. Un esempio, visibile nell'elenco a seguire, è l'ematocrito che viene calcolato servendosi del valore della concentrazione di emoglobina (che è misurata tramite metodo ottico).

Ecco ora un elenco di riferimento brevemente descrittivo dei parametri che verranno incontrati in seguito nei prossimi capitoli e visibili nella Figure 5.1, 5.2:

Anion gap (k+): Calcolato in mmol/L, è la differenza tra la concentrazione di ioni potassio (k^+) e sodio (Na^+) e la concentrazione di ioni cloro (Cl^-) e bicarbonato (HCO_3^-) ovvero $cK^+ + cNa^+ - cCl^- - cHCO_3^-(P)$, con $cHCO_3^-(P)$ che corrisponde alla concentrazione di bicarbonato nel plasma.

Anion gap: Calcolato in mmol/L, è la differenza tra la concentrazione di ioni sodio (Na^+) e la concentrazione di ioni cloro (Cl^-) e bicarbonato (HCO_3^-) ovvero $cNa^+ - cCl^- - cHCO_3^-(P)$, con $cHCO_3^-(P)$ che corrisponde alla concentrazione di bicarbonato nel plasma.

B: Misurata in millimetri di mercurio (mmHg) con un range che va dai 450 ai 800 mmHg, è la pressione barometrica misurata dal barometro incorporato nell'analizzatore in modo da compensare gli effetti di quest'ultima sui campioni di sangue.

Ca++: Indica la concentrazione di ioni calcio nel plasma. Misurato tramite metodo potenziometrico con un elettrodo dedicato, è espresso in mmol/L con un range che va da 0,20 a 9,99 mmol/L.

Ca(7.4): Rappresenta la concentrazione di ioni di calcio nel plasma ad un pH di 7.40 Calcolato dall'apparecchio tramite la seguente formula $cCa^{2+}(7.4) = cCa^{2+}[1 - 0.53 \times (7.40 - pH)]$ ed espresso in mg/dL

cBase(B) o ABE: L'Eccesso Attuale di Base, ovvero la concentrazione di base titolabile quando il sangue è titolato con una base forte o un acido, con un pH del plasma di 7,40, ad una pCO_2 di 5,33kPa e 37 ˚, all'attuale saturazione di ossigeno. Valori positivi (eccesso di base) indicano un relativo deficit di acidi non carbonici mentre valori negativi (deficit di base) segnalano un relativo eccesso di acidi non carbonici. Viene espresso in mmol/L e calcolato dall'apparecchio in base ai valori di pH, ctHb e pCO_2 .

cBase(ecf) o SBE: L'Eccesso Standard di Base, ovvero un'espressione "in vivo" dell'eccesso di base. Fa riferimento ad un modello del fluido extracellulare (una parte di sangue diluita da due parti di plasma) ed è calcolato usando un valore standard della concentrazione di emoglobina del fluido extracellulare totale. Calcolato in base al valore di cBase(B) e ctHb ed espresso in mmol/L.

CBase(Ecf,ox): Rappresenta la cBase(ecf) del sangue pienamente ossigenato. Calcolato in base ai valori di ctHb, cBase(B) e sO_2 ed espresso in mmol/L.

CL-: Indica la concentrazione di ioni cloruro nel plasma. Misurato tramite metodo potenziometrico con un elettrodo dedicato, è espresso in mmol/L con un range che va da 7 a 350 mmol/L.

COHB%: Indica la frazione di carbossiemoglobina nell'emoglobina totale nel sangue. Misurato tramite metodo ottico ed espresso in valore percentuale da 0 a 100%.

GLU: La concentrazione di D-glucosio nel plasma. Misurato tramite metodo amperometrico con un apposito elettrodo di argento e platino, è espresso in mg/dL con un range di misura che va da 0 a 1081 mg/dL.

HBF%: Indica la frazione di emoglobina fetale nell'emoglobina totale nel sangue. Misurato tramite metodo ottico ed espresso in valore percentuale da 0 a 100%.

HCO3-: Concentrazione di bicarbonato nel plasma. Calcolato dall'apparecchio usando i valori misurati di pH e pCO_2 ed espresso in mmol/L.

HCO3-(st) o SBC: Bicarbonato Standard, ovvero la concentrazione del carbonato di idrogeno nel plasma del sangue che è stato equilibrato con una mistura di gas con pCO_2=5,33kPa e PO2≥13,33kPa a 37˚C. Calcolato in base ai valori di ctHb, cBase(B) e sO_2 ed espresso in mmol/L.

HCT: Indica l'ematocrito, ovvero il rapporto tra il volume di eritrociti e il volume di sangue. Calcolato in base al valore di ctHb ed espresso in percentuale.

K+: Indica la concentrazione di ioni potassio nel plasma. Misurato tramite metodo potenziometrico con un elettrodo dedicato, è espresso in mmol/L con un range che va da 0,5 a 25,0 mmol/L.

LAC: La concentrazione di L-lattato nel plasma. Misurato tramite metodo amperometrico con un apposito elettrodo di argento e platino, è espresso in mmol/L con un range di misura che va da 0 a 30 mmol/L.

METHB%: Indica la frazione di metemoglobina nell'emoglobina totale nel sangue. Misurato tramite metodo ottico ed espresso in valore percentuale da 0 a 100%.

mOsm: Osmolalità del plasma. Calcolata servendosi dei valori di cNa^+ e cGlu ed espresso in mmol/kg

NA+: Indica la concentrazione di ioni sodio nel plasma. Misurato tramite metodo potenziometrico con un elettrodo dedicato, è espresso in mmol/L con un range che va da 7 a 350 mmol/L.

O2HB: Indica la frazione di ossiemoglobina sull'emoglobina totale nel sangue. Misurato tramite metodo ottico ed espresso in valore percentuale da 0 a 100%.

P50(st): Pressione parziale di ossigeno a metà saturazione (50%) nel sangue in condizioni standard quali T=37˚C pH=7,40 pCO2=5,33kPa FCOHb, FMetHb, FHbF = 0

PCO2: La pressione parziale del biossido di carbonio (anidride carbonica) nel sangue, un suo valore alto o basso nel sangue arterioso indica rispettivamente ipercapnia e ipocapnia. Misurato tramite metodo potenziometrico con un elettrodo dedicato, è espresso in mmHg con un range che va da 5,0 a 99,9 mmHg.

PH: Indica l'acidità o l'alcalinità del campione di sangue. Misurato tramite metodo potenziometrico con un elettrodo dedicato con un range che va da 6,300 a 8,000.

PO2: Indica la pressione parziale di ossigeno nel sangue, valori alti o bassi indicano rispettivamente iperossia e ipossia. Misurato tramite metodo amperometrico con un elettrodo dedicato, è espresso in mmHg con un range che va da 0,0 a 99,9 mmHg.

RHB% o *F*HHb: La frazione di deossiemoglobina sull'emoglobina totale nel sangue. Misurato tramite metodo ottico ed espresso in valore percentuale da 0 a 100%.

SO2: La saturazione di ossigeno, ovvero il rapporto tra la concentrazione di ossiemoglobina e l'emoglobina meno le emoglobine non portatrici di ossigeno (carbossiemoglobina e metemoglobina). Misurato tramite metodo ottico ed espresso in valore percentuale da 0 a 100%.

tBil: Indica la concentrazione di bilirubina totale nel plasma, comprendendo sia la sua forma coniugata che la non coniugata. Misurato tramite metodo ottico ed espresso in mg/dL con un range che va da 0,0 a 58,5 mg/dL.

tCO2(B)%: Indica la concentrazione di anidride carbonica totale nel sangue calcolata basandosi sulle concentrazioni di CO2 totali nel plasma e nel liquido eritrocitario.

THB: (ctHb) La concentrazione di emoglobina totale nel sangue. Misurato tramite metodo ottico ed espresso in g/dL con un range che va da 0,0 a 27,7 g/dL

A parte vanno menzionati poi i parametri né misurati né calcolati, ovvero i parametri *immessi* direttamente dall'operatore, come ad esempio nome e codice ospedaliero del paziente.

Di seguito un breve elenco dei parametri immessi che verranno menzionati nei prossimi capitoli:

Codice: Il codice identificativo del paziente.

Data analisi: La data dell'analisi.

FIO2%: Frazione di ossigeno nell'aria secca inspirata, immesso dall'operatore in valore percentuale.

mod. di misura: Contiene valori come C 125uL, C 35uL MET, S 250uL. Indica cosa è stato usato per il prelievo e relativa capacità volumetrica: S per siringhe e C per i capillari.

Prelievo: L'operatore segnala che tipo di prelievo è stato effettuato. I tipi di prelievo in queste emogasanalisi sono: Aria espirata, arteria ombelicale, arteria radiale, arteria tibiale, capillare arterializzato, capillare venoso, cordone arterioso, cordone misto, cordone venoso, liquor, urine, vena centrale, vena periferica.

RQ: Quoziente respiratorio, ovvero il rapporto tra la produzione di CO_2 e il consumo di O_2.

Tipo: L'operatore immette quale tipo di test vuole effettuare, sono: arterioso, capillare, cordone ombelicale, scalpo fetale, venoso.

3 PROCESSAMENTO DEI DATI

In questo capitolo esporrò come si presentavano i dati prima dell'elaborazione e come si presentano a lavoro concluso.

3.1 I DATI

I dati su cui è stato svolto questo lavoro sono i risultati delle emogasanalisi, d'ora in poi abbreviate in **EGA**, sotto forma di 19 file Excel (.xls) separati. Essi comprendono 9 anni di prelievi, dal 13/11/2008 al 09/11/2017 per un totale di 86˙602 emogasanalisi.

Paziente	Codice	Analizzatore	Data Analisi	Tipo	PH	PCO2 mmHg	PO2 mmHg	HCO3- mmo/L	SBC mmol/L	tCO2(B) Vol%	ABE mmol/L	SBE mmol/L	cBase(Ecf,ox) mmol/l	THB g/dL	HCT %	SO2 %	RHB %	METHB %	HBF %	COHB %	N m
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare		(722) - Err	(722) - Err	4,47		21,2		-2,8		(722) - Err		(722) - Err	(722) - Err	(722) - Err	(722) - Err	(722) - Err	(7
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,334	47,1	47,4	24,4	22,8	46,4	-1,6	-0,7	-0,8	18,0	55,0	88,7	11,1	0,6	91	1,0	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare		(722) - Err	(722) - Err	4,47		21,2		-2,8		(722) - Err		(722) - Err	(722) - Err	(722) - Err	(722) - Err	(722) - Err	(7
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,334	47,1	47,4	24,4	22,8	46,4	-1,6	-0,7	-0,8	18,0	55,0	88,7	11,1	0,6	91	1,0	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,382	41,2	50,3	24,0	23,7	45,8	-0,6	-0,5	-0,6	16,8	51,3	90,3	9,5	0,8	89	1,3	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,382	43,6	54,6	25,3	24,6	47,2	0,4	0,8	0,8	18,5	56,4	93,5	6,4	0,5	91	1,7	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,337	53,3	41,8	27,8	24,8	52,7	1,0	2,5	2,4	18,5	56,6	84,7	15,0	0,4	91	1,6	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,382	41,2	50,3	24,0	23,7	45,8	-0,6	-0,5	-0,6	16,8	51,3	90,3	9,5	0,8	89	1,3	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,382	43,6	54,6	25,3	24,6	47,2	0,4	0,8	0,8	18,5	56,4	93,5	6,4	0,5	91	1,7	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,337	53,3	41,8	27,8	24,8	52,7	1,0	2,5	2,4	18,5	56,6	84,7	15,0	0,4	91	1,6	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Venoso	7,325	47,6	20,9	24,1	21,6	48,8	-1,8	-1,1	-1,6	14,9	45,8	42,7	56,6	1,0		0,3	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Venoso	7,325	47,6	20,9	24,1	21,6	48,8	-1,8	-1,1	-1,6	14,9	45,8	42,7	56,6	1,0		0,3	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Venoso	7,339	44,6	24,9	23,3	21,5	46,4	-2,1	-1,6	-2,0	15,8	48,4	53,0	46,3	0,8	88	0,6	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Venoso	7,339	44,6	24,9	23,3	21,5	46,4	-2,1	-1,6	-2,0	15,8	48,4	53,0	46,3	0,8	88	0,6	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,404	36,7	48,6	22,5	23,3	41,1	-1,1	-1,5	-1,6	19,4	59,3	91,8	8,0	0,5	88	1,4	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,404	36,7	48,6	22,5	23,3	41,1	-1,1	-1,5	-1,6	19,4	59,3	91,8	8,0	0,5	88	1,4	1:
[illegible]	[illegible]	ABL800 A	[illegible]	Capillare	7,442	32,0	51,1	21,5	23,5	38,8	-0,9	-2,0	-2,1	19,5	59,4	92,5	7,3	0,5	84	1,6	1:

Fig. 5.1 - Esempio di file Excel di vari EGA-prima metà

%	NA+ mmol/L	K+ mmol/L	CA++ mmol/dL	Ca(7.4) mmol/dL	CL- mmol/L	GLU mmol/dL	LAC mmol/L	tBil mmol/dL	Anion gap (K+) mmol/L	ANION GAP mmol/L	mOsm mmol/Kg	CREA mg/dl	T ° Cel	FIO2 %	PRELIEVO	P50(st) mmHg	RQ	O2HB %	B mmHg	cBase(B) mmol/L	Mod. di misura
:rr	(722) - Err	(722) - Err	(722) - Err	-2,8	(722) - Err	(722) - Err	77					2,3	(719) - Ca			(722) - Err		(722) - Err	22,0		C 125uL
	127	6,0	5,15	4,97	110	74	2,8	1,8	-1,2		257,5	0,90	37,0	23,0	CAPILLAR	26,85	0,86	87,3	748	-1,6	C 125uL
:rr	(722) - Err	(722) - Err	(722) - Err	-2,8	(722) - Err	(722) - Err	77					2,3	(719) - Ca			(722) - Err		(722) - Err	22,0		C 125uL
	127	6,0	5,15	4,97	110	74	2,8	1,8	-1,2		257,5	0,90	37,0	23,0	CAPILLAR	26,85	0,86	87,3	748	-1,6	C 125uL
	135	4,4	5,02	4,97	114	87	1,9	7,5	1,0		274,9	0,84	37,0	21,0	CAPILLAR	26,85	0,86	88,4	758	-0,6	C 125uL
	134	5,2	4,85	4,80	117	85	2,0	8,7	-3,2		272,8	0,77	37,0	21,0	CAPILLAR	26,85	0,86	91,4	752	0,4	C 125uL
	139	4,7	4,79	4,63	118	68	1,2	8,7	-2,7		280,8	0,87	36,7	21,0	CAPILLAR	26,85	0,86	83,0	748	1,0	C 125uL
	135	4,4	5,02	4,97	114	87	1,9	7,5	1,0		274,9	0,84	37,0	21,0	CAPILLAR	26,85	0,86	88,4	758	-0,6	C 125uL
	134	5,2	4,85	4,80	117	85	2,0	8,7	-3,2		272,8	0,77	37,0	21,0	CAPILLAR	26,85	0,86	91,4	752	0,4	C 125uL
	139	4,7	4,79	4,63	118	68	1,2	8,7	-2,7		280,8	0,87	36,7	21,0	CAPILLAR	26,85	0,86	83,0	748	1,0	C 125uL
	132	3,9	4,85	4,66	114	47	2,8	0,4	-2,1		266,0	0,65	37,0	21,0	CORDONE	26,85	0,86	42,1	753	-1,8	S 250uL
	132	3,9	4,85	4,66	114	47	2,8	0,4	-2,1		266,0	0,65	37,0	21,0	CORDONE	26,85	0,86	42,1	753	-1,8	S 250uL
	132	3,5	5,05	4,89	114	48	2,6	-0,0	-1,9		266,4	0,69	37,0	21,0	CORDONE	26,85	0,86	52,3	753	-2,1	S 250uL
	132	3,5	5,05	4,89	114	48	2,6	-0,0	-1,9		266,4	0,69	37,0	21,0	CORDONE	26,85	0,86	52,3	753	-2,1	S 250uL
	138	5,1	5,02	5,03	119	121	1,4	14,7	1,4		282,6	0,43	36,0	21,0	CAPILLAR	26,85	0,86	90,1	764	-1,1	C 125uL
	138	5,1	5,02	5,03	119	121	1,4	14,7	1,4		282,6	0,43	36,0	21,0	CAPILLAR	26,85	0,86	90,1	764	-1,1	C 125uL
	136	5,2	4,87	4,98	118	105	1,3	15,3	0,9		277,1	0,41	37,0	21,0	CAPILLAR	26,85	0,86	90,6	764	-0,9	C 125uL
	136	5,2	4,87	4,98	118	105	1,3	15,3	0,9		277,1	0,41	37,0	21,0	CAPILLAR	26,85	0,86	90,6	764	-0,9	C 125uL
:rr	(722) - Err	(722) - Err	(722) - Err		4,02	114	35					2,0	(721) - Ca			?		(722) - Err		?	S 250uL
:rr	(722) - Err	(722) - Err	(722) - Err		4,02	114	35					2,0	(721) - Ca			?		(722) - Err		?	S 250uL

Fig. 5.2 - Esempio di file Excel di vari EGA-seconda metà

Le figure 5.1 e 5.2 sono un esempio (Fig. 5.2 da affiancare a destra di Fig.5.1 poiché è il suo continuo) di come si presentano i dati del periodo 2008-2010 in origine nel formato Excel: ogni riga corrisponde ad un EGA mentre le colonne contengono i parametri, con i valori di "Data analisi", "Codice" e "Paziente" oscurati per questioni di privacy. Si può notare come i dati sono composti da caratteri stringa ("Tipo", "Prelievo", "Paziente"), numerici, misti (i messaggi di errore e "Mod. di misura") e campi vuoti.

Ad una prima analisi si è reso subito evidente il fatto che il nome, il numero di parametri e il formato in cui sono inseriti nei file varia a seconda del periodo in cui è stato effettuato l'esame: 42 parametri nel periodo 2008-2010, 41 tra il 2011-2014 e 36 nel 2015-2017, una differenza che può indicare che alcuni parametri si ripetono più volte nello stesso EGA (doppioni) o che alcuni parametri sono assenti del tutto.

Paziente	Paziente	Paziente
Codice	Codice	Codice
Analizzatore	Analizzatore	Analizzatore
Data Analisi	Data Analisi	Data Analisi
Tipo	Tipo	Tipo
PH	PH	PH
PCO2 mmHg	PCO2 mmHg	PCO2 mmHg
PO2 mmHg	PO2 mmHg	PO2 mmHg
HCO3-mmo/L	HCO3- mmo/L	HCO3- mmo/L
SBC mmol/L	HCO3- (st) mmo/L	SBC mmol/L
tCO2(B) Vol%	HCO3-(st) mmo/L	tCO2(B) Vol%
ABE mmol/L	SBC mmol/L	ABE mmol/L
SBE mmol/L	tCO2(B) Vol%	SBE mmol/L
cBase(Ecf,ox) mmol/l	ABE mmol/L	cBase(Ecf,ox) mmol/l
THB g/dL	SBE mmol/L	THB g/dL
HCT %	cBase(Ecf,ox) mmol/l	HCT %
SO2 %	THB g/dL	SO2 %
RHB %	HCT %	RHB %
METHB %	SO2 %	METHB %
HBF %	RHB %	HBF %
COHB %	METHB %	COHB %
NA+ mmol/L	HBF %	NA+ mmol/L
K+ mmol/L	COHB %	K+ mmol/L
CA++ mg/dL	NA+ mmol/L	CA++ mg/dL
Ca(7.4) mg/dL	K+ mmol/L	Ca(7.4) mg/dL
CL- mmol/L	CA++ mg/dL	CL- mmol/L
GLU mg/dL	Ca(7.4) mg/dL	GLU mg/dL
LAC mmol/L	CL- mmol/L	LAC mmol/L
tBil mg/dL	GLU mg/dL	tBil mg/dL
Anion gap (K+) mmol/L	LAC mmol/L	Anion gap (K+) mmol/L
ANION GAP mmol/L	tBil mg/dL	ANION GAP mmol/L
mOsm mmol/Kg	Anion gap (K+) mmol/L	mOsm mmol/Kg
CREA mg/dl	ANION GAP mmol/L	CREA mg/dl
T ° Cel	mOsm mmol/Kg	T ° Cel
FIO2 %	CREA mg/dl	FIO2 %
PRELIEVO	T ° Cel	PRELIEVO
P50(st) mmHg	FIO2 %	
RQ	PRELIEVO	
O2HB %	Ca2+ mg/dL	
B mmHg	GAP (K+)	
cBase(B) mmol/L	O2HB	
Mod. di misura		

Fig. 6 - Confronto dei parametri nei tre periodi

Nella Fig. 6 si può osservare ciò che si è detto poco fa. I parametri che nelle FIG. EXCEL occupavano la riga di testa sono stati riorientati in verticale per permettere un confronto più immediato, con il periodo 2008-2010 in rosso, il 2011-2014 in blu e il 2015-2017 in giallo. Il periodo2008-2010 e il 2015-2017 presentano il maggior grado di specularità con l'unica differenza che l'ultimo non ha gli ultimi sei parametri che nel primo seguono il parametro "PRELIEVO". Il periodo 2011-2014 è quello che più si discosta dagli altri per l'ordine in cui sono presentati i parametri ed il modo in cui alcuni di loro sono stati scritti: i formati sono identici fino al nono parametro "HCO3-" poi il primo prosegue con "SBC" mentre il secondo inserisce invece "HCO3-(st)" in doppia copia, ripetizione che avviene anche nel caso di "Anion gap(k+)" che viene ripetuto con "GAP(K+)", inoltre la concentrazione di ioni calcio viene scritta come Ca2+ invece di Ca++. Considerazione finale di questa iniziale esposizione dei dati è che se si considera che ogni EGA presenta dai 36 ai 42 parametri coi rispettivi valori e le EGA presenti sono 86602, ne risulta che i dati grezzi totali ammontano a più di tre milioni di valori.

3.2 PULIZIA DEI DATI

Visto il modo in cui si presentavano i dati il primo e più importante step (e anche uno dei principali obbiettivi di questo lavoro) è stato quello di riarrangiare tutti i dati in un formato unico, processabile in Matlab e capace di gestire una grande mole di dati, in modo da permettere non solo l'elaborazione di tali dati nell'ambiente Matlab ma anche di farlo in maniera veloce ed efficiente, andando a costituire una solida base per qualsiasi operazione che si voglia svolgere su questi dati.

Il primo passo è stato quello di individuare i cosiddetti parametri doppione, ovvero quei parametri con un nome diverso ma che stanno ad indicare la stessa cosa, come ad esempio ABE e cBase(B). Questi doppioni sono presenti sia come parametro ridondante ripetuto in una stessa EGA sia come differente notazione tra gli EGA dei tre periodi di cui al paragrafo precedente.

ABE mmol/L	cBase(B) mmol/L	
Anion gap (K+) mmol/L	GAP (K+)	
ANION GAP mmol/L		
B mmHg		
Ca(7.4) mg/dL		
CA++ mg/dL	Ca2+ mg/dL	
cBase(Ecf,ox) mmol/l		
CL- mmol/L		
Codice		
COHB %		
CREA mg/dl		
Data Analisi		
FIO2 %		
GLU mg/dL		
HBF %		
HCO3- (st) mmo/L	HCO3-(st) mmo/L	SBC mmol/L
HCO3- mmo/L		
HCT %		
K+ mmol/L		
LAC mmol/L		

METHB %	
Mod. di misura	
mOsm mmol/Kg	
NA+ mmol/L	
O2HB	O2HB %
P50(st) mmHg	
PCO2 mmHg	
PH	
PO2 mmHg	
PRELIEVO	
RHB %	
RQ	
SBE mmol/L	
SO2 %	
T ° Cel	
tBil mg/dL	
tCO2(B) Vol%	
THB g/dL	
Tipo	

Fig. 7 - Elenco dei cinque parametri con i rispettivi doppioni.

Dopo questa iniziale operazione ne segue la creazione dello script principale in ambiente Matlab, denominato "*struttura*".[1] (Cod. 1.1-1.5). Ho creato "*struttura*" come una funzione matlab, ovvero uno script che può essere richiamato in qualsiasi momento usando la giusta sintassi di: nome della funzione, input e output. In questo caso "*struttura*", come si può vedere in Cod. 1.1, prende in ingresso un file EGA in formato .xls e crea due strutture: una struttura contenente i messaggi di errore dell'apparecchiatura e relativa appartenenza e un'altra struttura contenente i dati nella nuova formattazione. Una struttura Matlab (*struct*) è un tipo di dato formato da più elementi, detti campi. Ciascun campo è identificato da un nome e può essere un qualsiasi tipo di dato Matlab come ad esempio scalari o array di caratteri.

Facendo riferimento al codice visibile in Cod. 1.1, 1.2, 1.3 ho per prima cosa inizializzato la struttura e, avendo impostato un array "nomi" contenente tutti i parametri da ricercare, ho creato un breve script che cerca in quale colonna del file Excel si trova il c-esimo parametro contenuto nell'array "nomi". C-esimo poiché la ricerca avviene tramite un principale ciclo for per c che va da 1 alla lunghezza dell'array "nomi", ovvero cerca il nome dei parametri in ordine partendo dal primo elemento dell'array "nomi", in questo caso "Codice", fino ad arrivare all'ultimo, e cioè "Tipo". Per fare un esempio poniamo che il ciclo for sia arrivato a c=3: il terzo valore dell'array "nomi" è "ABE mmol/L", la funzione *strfind* confronta "ABE mmol/L" con i nomi dei parametri presenti nel file Excel e viene individuata la dodicesima colonna del file Excel. A questo punto viene creato il campo della struttura e riempito coi valori della colonna Excel corrispondente (convertendo i valori da Excel a Matlab tramite la funzione *cell2mat*), col campo in questione che prende il nome del parametro ricercato. Nel caso in cui il parametro non venga trovato nel file Excel il campo della

[1] Si presenta purtroppo un caso di omonimia tra la funzione da me creata, *struttura*, e la "struttura" come tipo di dato Matlab. La struttura come funzione verrà espressa in corsivo.

struttura viene creato ugualmente ma verrà riempito di NaN (Not A Number). A questo punto la struttura è una copia esatta del file Excel, (fatta eccezione per i nomi di alcuni parametri rinominati in maniera opportuna) in un formato più facilmente gestibile nell'ambiente Matlab.

Arrivato a questo punto i dati presentano ancora due problemi principali: sono ancora presenti i parametri doppione e i dati non sono ancora pienamente leggibili in matlab.

Durante l'elaborazione è infatti emerso che i campi che avrebbero dovuto contenere solo valori numerici contenevano caratteri non numerici, messaggi di errore e campi vuoti: tutti fattori che impedivano a Matlab di operare sui dati. Difatti, nel caso si voglia usare una funzione o operazione che richiede e quindi si aspetta tutti dati di uno stesso tipo (numerici, array di caratteri, array di stringhe, etc.), la presenza di una variabile di un altro tipo causa un errore della funzione e il conseguente fallimento dell'operazione desiderata.

Nel codice visibile in Cod. 1.4 ho rimediato a questo problema in un procedimento a due fasi: ricerca e risoluzione.

Per prima cosa ho cercato i valori non numerici, escludendo dalla ricerca i campi, e cioè i parametri, che fosse naturale contenessero sia numeri che lettere come "Codice", "Data", "Mod di misura", "PRELIEVO" e "Tipo". Esaminando il resto dei campi ho rilevato la presenza di:

- Virgole come separatrici decimali.
- Punti interrogativi.
- Cinque puntini di sospensione al posto dei valori numerici.
- Campi vuoti.
- Messaggi di errore.

Ognuna di queste problematiche (visibili anche già nelle Fig. 5.1, 5.2) ha richiesto un approccio apposito (Cod. 1.4) in base al tipo di problema che rappresentavano in relazione a come Matlab gestisce i tipi di dati, anche se per la maggior parte dei casi è stata usata la funzione *strrep.* L'uso della virgola come separatrice decimale non è supportato da Matlab che richiede infatti il punto, per cui tramite una serie di cicli *for* e la funzione *strrep* ho sostituito tutti le virgole con punti. I punti interrogativi "?" sono stati sostituiti anche in questo caso tramite iterazioni di *strrep* sostituendoli con il carattere vuoto " " , dopodiché questi nuovi campi vuoti e i già presenti campi vuoti sono stati sostituiti con il valore NaN tramite l'utilizzo di cicli for e la funzione *isempty,* una funzione logica che identifica facilmente i campi vuoti poiché restituisce 1 o 0 in base al fatto che l'array sia pieno o vuoto. Ho dovuto agire in maniera diversa sui puntini di sospensione perché andavano rimossi senza essere confusi con i regolari punti di separazione decimale, per cui tramite un'iterazione di *strrep* li ho inizialmente sostituiti con il carattere "E" per poi sostituirli col NaN nel processo di eliminazione dei messaggi di errore, spiegato a seguito. I messaggi di errore rappresentavano un ostacolo al processamento dei dati visto che erano array di stringhe presenti là dove volevo rimanessero solo i valori numerici ma costituiscono purtuttavia delle informazioni utili la cui eliminazione andava evitata, a questo scopo ho creato il secondo output della funzione *"struttura":* la struttura "ERRORE". Come visibile in Cod. 1.4, il codice, tramite la funzione *isstrprop*, setaccia i vari campi della struttura alla ricerca di caratteri alfabetici, escludendo dalla ricerca i campi in cui è normale ci siano tali caratteri, e gli unici caratteri alfabetici rimasti sono proprio i messaggi di errore e i "….." convertiti in "E". A questo punto i messaggi di errore vengono inseriti nel campo "errore" della struttura "ERRORE", affiancati dai campi "riga" e "field" che contengono rispettivamente in

quale riga e campo della struttura era presente il messaggio di errore; nella struttura dati i messaggi di errore e le “E” vengono infine rimpiazzati da NaN.

Finito questo processo la struttura dati presenta solo caratteri numerici e NaN nei campi in cui deve, tuttavia visto che i dati provengono dal formato Excel, Matlab li interpreta ancora come stringhe ed è stato necessario applicare estensivamente la funzione *str2num* per trasformare i dati in array numerici per permettere il *plotting*[2] dei parametri.

Fields	riga	field	errore
222	171	'NA'	'(476) - Misurazione instabile'
223	171	'T_Cel'	'(712) - Misurazione FHb non possibile'
224	172	'T_Cel'	'(712) - Misurazione FHb non possibile'
225	178	'GLU'	'(210) - Errori in calibrazione presenti'
226	182	'GLU'	'(210) - Errori in calibrazione presenti'
227	185	'T_Cel'	'(712) - Misurazione FHb non possibile'
228	186	'T_Cel'	'(712) - Misurazione FHb non possibile'
229	189	'T_Cel'	'(712) - Misurazione FHb non possibile'
230	190	'T_Cel'	'(712) - Misurazione FHb non possibile'
231	197	'COHB'	'E'
232	197	'METHB'	'E'
233	197	'O2HB'	'E'
234	197	'Doppione_6'	'E'

Fig. 8 – Esempio della struttura ERRORE: ogni riga della struttura contiene campo e riga in cui era presente il relativo messaggio di errore.

L’ultima operazione della funzione *struttura* è quello di eliminare i parametri doppioni, lasciando una struttura definitiva a 39 parametri unici. Come si può vedere in Cod 1.5, i parametri doppione sono già stati identificati nel lavoro precedente e la variabile “seldop” contiene, in maniere ordinata, i cinque parametri con i relativi parametri doppione rinominati:

- ABE e Doppione_1 (in precedenza cBase(B))
- ANION_GAP_K e Doppione_2 (in precedenza GAP(K+))
- CA_2piu e Doppione_3 (in precedenza Ca2+)
- HCO3_st con Doppione_4 e Doppione_5 (in precedenza HCO3-(st) e SBC)
- O2HB e Doppione_6 (in precedenza O2HB%)

Lo script analizza tutto il contenuto del campo doppione e se esso contiene valori essi vengono trasferiti nel campo col nome del parametro e il campo doppione viene eliminato, se il campo doppione non contiene dati viene invece direttamente eliminato. Per esempio, se il campo Doppione_1 ha tutti NaN viene rimosso dai campi della struttura, se invece ha dei dati al suo interno questi dati vengono sovrascritti in ABE e poi Doppione_1 viene eliminato dai campi della struttura. Da notare è che se entrambi i campi presentano dati, la sovrascrittura di uno dei campi non comporta una perdita dei dati perché entrambi i campi contengono gli stessi valori.

La funzione *struttura* fin qui esposta viene poi utilizzata nello script in Cod. 2 per creare una struttura completa contenente i dati EGA di tutti i file Excel. La funzione *struttura* infatti lavora un solo file .xls alla volta ma inserendola in un ciclo for e fornendole i nomi di tutti i file Excel che si vuole processare essa creerà un'unica struttura dati e un'unica struttura degli errori. L’input a struttura è automatizzato, richiedendo solamente il percorso della cartella dove sono contenuti i file EGA: i nomi dei file contenuti nella cartella vengono letti dalla funzione *dir* e poi da *fieldnames*, poiché *dir* crea una struttura che contiene i nomi dei file nel suo terzo campo. La struttura di partenza viene inizializzata esternamente al ciclo for col primo file in modo da avere già una struttura dati ed errore su cui il ciclo for possa lavorare, uno degli input della funzione

[2] Per plotting si intende la rappresentazione dei dati tramite un grafico in Matlab.

struttura è la lunghezza della struttura precedente in modo che la posizione degli errori nella struttura errore sia corretta in riferimento alla struttura errori completa.

Utilizzando questi 2 script insieme si è venuta a creare una struttura dati completa di 86602 EGA con 39 parametri unici, i cui valori numerici sono rappresentabili tramite grafico con la funzione *plot* e velocemente gestibili con funzioni Matlab.

Variables - x

x

1x86602 struct with 39 fields

Fields	Codice	Data	ABE	ANION_GAP_K	ANION_GAF	B_mmHg	Ca_7_4	CA_2piu	cBase_Ecf_o>	CL	COHB	CREA	FIO2	GLU	HBF	HCO3_st	HCO3	HCT	K	LAC
1	[illegible]	[illegible]	NaN	NaN	NaN	22	-2.8000	NaN	NaN	NaN	NaN	2.3000	NaN	NaN	NaN	NaN	4.4700	NaN	NaN	[illegible]
2	[illegible]	[illegible]	-1.6000	-1.2000	NaN	748	4.9700	5.1500	-0.8000	110	1	0.9000	23	74	91	22.8000	24.4000	55	6	2.80
3	[illegible]	[illegible]	NaN	NaN	NaN	22	-2.8000	NaN	NaN	NaN	NaN	2.3000	NaN	NaN	NaN	NaN	4.4700	NaN	NaN	[illegible]
4	[illegible]	[illegible]	-1.6000	-1.2000	NaN	748	4.9700	5.1500	-0.8000	110	1	0.9000	23	74	91	22.8000	24.4000	55	6	2.80
5	[illegible]	[illegible]	-0.6000	1	NaN	758	4.9700	5.0200	-0.6000	114	1.3000	0.8400	21	87	89	23.7000	24	51.3000	4.4000	1.90
6	[illegible]	[illegible]	0.4000	-3.2000	NaN	752	4.8000	4.8500	0.8000	117	1.7000	0.7700	21	85	91	24.6000	25.3000	56.4000	5.2000	
7	[illegible]	[illegible]	1	-2.7000	NaN	748	4.6300	4.7900	2.4000	118	1.6000	0.8700	21	68	91	24.8000	27.8000	56.6000	4.7000	1.20
8	[illegible]	[illegible]	-0.6000	1	NaN	758	4.9700	5.0200	-0.6000	114	1.3000	0.8400	21	87	89	23.7000	24	51.3000	4.4000	1.90
9	[illegible]	[illegible]	0.4000	-3.2000	NaN	752	4.8000	4.8500	0.8000	117	1.7000	0.7700	21	85	91	24.6000	25.3000	56.4000	5.2000	
10	[illegible]	[illegible]	1	-2.7000	NaN	748	4.6300	4.7900	2.4000	118	1.6000	0.8700	21	68	91	24.8000	27.8000	56.6000	4.7000	1.20
11	[illegible]	[illegible]	-1.8000	-2.1000	NaN	753	4.6600	4.8500	-1.6000	114	0.3000	0.6500	21	47	NaN	21.6000	24.1000	45.8000	3.9000	2.80
12	[illegible]	[illegible]	-1.8000	-2.1000	NaN	753	4.6600	4.8500	-1.6000	114	0.3000	0.6500	21	47	NaN	21.6000	24.1000	45.8000	3.9000	2.80
13	[illegible]	[illegible]	-2.1000	-1.9000	NaN	753	4.8900	5.0500	-2	114	0.6000	0.6900	21	48	88	21.5000	23.3000	48.4000	3.5000	2.60
14	[illegible]	[illegible]	-2.1000	-1.9000	NaN	753	4.8900	5.0500	-2	114	0.6000	0.6900	21	48	88	21.5000	23.3000	48.4000	3.5000	2.60
15	[illegible]	[illegible]	-1.1000	1.4000	NaN	764	5.0300	5.0200	-1.6000	119	1.4000	0.4300	21	121	88	23.3000	22.5000	59.3000	5.1000	1.40
16	[illegible]	[illegible]	-1.1000	1.4000	NaN	764	5.0300	5.0200	-1.6000	119	1.4000	0.4300	21	121	88	23.3000	22.5000	59.3000	5.1000	1.40
17	[illegible]	[illegible]	-0.9000	0.9000	NaN	764	4.9800	4.8700	-2.1000	118	1.6000	0.4100	21	105	84	23.5000	21.5000	59.4000	5.2000	1.30
18	[illegible]	[illegible]	-0.9000	0.9000	NaN	764	4.9800	4.8700	-2.1000	118	1.6000	0.4100	21	105	84	23.5000	21.5000	59.4000	5.2000	1.30
19	[illegible]	[illegible]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0200	NaN	2	NaN	114	NaN	NaN	NaN	NaN	NaN	[illegible]
20	[illegible]	[illegible]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0200	NaN	2	NaN	114	NaN	NaN	NaN	NaN	NaN	[illegible]
21	[illegible]	[illegible]	1.3000	-0.8000	NaN	751	5.2900	5.4300	2.4000	111	0.8000	0.7100	21	117	98	25.2000	27.6000	52.2000	5.4000	2.40
22	[illegible]	[illegible]	1.3000	-0.8000	NaN	751	5.2900	5.4300	2.4000	111	0.8000	0.7100	21	117	98	25.2000	27.6000	52.2000	5.4000	2.40
23	[illegible]	[illegible]	1	3.9000	NaN	764	5.2100	5.3500	1.8000	110	1.3000	0.9800	21	67	92	25	27	48.2000	4.9000	1.90
24	[illegible]	[illegible]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.5000	NaN	21	NaN	95	NaN	NaN	NaN	8.2000	Na
25	[illegible]	[illegible]	-3.4000	-1.9000	NaN	751	4.8300	4.8700	-4	121	1.7000	1.0400	21	74	97	21.2000	20.5000	51	6.9000	3.50
26	[illegible]	[illegible]	NaN	NaN	NaN	24.1000	0.4000	NaN	NaN	NaN	NaN	2.6000	NaN	NaN	NaN	NaN	4.5400	NaN	NaN	[illegible]

Fig. 9.1 – La struttura completa con 39 parametri, ogni riga è un EGA.

Variables - x

x

1x86602 struct with 39 fields

Fields	LAC	METHB	mOsm	NA	O2HB	P50_st	PCO2	PH	PO2	RHB	RQ	SBE	SO2	T_Cel	tBil	tCO2_B	THB	Mod_di_misura	PRELIEVO	Tipo
1	77	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-2.8000	NaN	NaN	NaN	21.2000	NaN	'C 125uL'	NaN	'Capillare'
2	2.8000	0.6000	257.5000	127	87.3000	26.8500	47.1000	7.3340	47.4000	11.1000	0.8600	-0.7000	88.7000	37	1.8000	46.4000	18	'C 125uL'	'CAPILLARE ...	'Capillare'
3	77	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-2.8000	NaN	NaN	NaN	21.2000	NaN	'C 125uL'	NaN	'Capillare'
4	2.8000	0.6000	257.5000	127	87.3000	26.8500	47.1000	7.3340	47.4000	11.1000	0.8600	-0.7000	88.7000	37	1.8000	46.4000	18	'C 125uL'	'CAPILLARE ...	'Capillare'
5	1.9000	0.8000	274.9000	135	88.4000	26.8500	41.2000	7.3820	50.3000	9.5000	0.8600	-0.5000	90.3000	37	7.5000	45.8000	16.8000	'C 125uL'	'CAPILLARE ...	'Capillare'
6	2	0.5000	272.8000	134	91.4000	26.8500	43.6000	7.3820	54.6000	6.4000	0.8600	0.8000	93.5000	37	8.7000	47.2000	18.5000	'C 125uL'	'CAPILLARE ...	'Capillare'
7	1.2000	0.4000	280.8000	139	83	26.8500	53.3000	7.3370	41.8000	15	0.8600	2.5000	84.7000	36.7000	8.7000	52.7000	18.5000	'C 125uL'	'CAPILLARE ...	'Capillare'
8	1.9000	0.8000	274.9000	135	88.4000	26.8500	41.2000	7.3820	50.3000	9.5000	0.8600	-0.5000	90.3000	37	7.5000	45.8000	16.8000	'C 125uL'	'CAPILLARE ...	'Capillare'
9	2	0.5000	272.8000	134	91.4000	26.8500	43.6000	7.3820	54.6000	6.4000	0.8600	0.8000	93.5000	37	8.7000	47.2000	18.5000	'C 125uL'	'CAPILLARE ...	'Capillare'
10	1.2000	0.4000	280.8000	139	83	26.8500	53.3000	7.3370	41.8000	15	0.8600	2.5000	84.7000	36.7000	8.7000	52.7000	18.5000	'C 125uL'	'CAPILLARE ...	'Capillare'
11	2.8000	1	266	132	42.1000	26.8500	47.6000	7.3250	20.9000	56.6000	0.8600	-1.1000	42.7000	37	0.4000	48.8000	14.9000	'S 250uL'	'CORDONE ...	'Venoso'
12	2.8000	1	266	132	42.1000	26.8500	47.6000	7.3250	20.9000	56.6000	0.8600	-1.1000	42.7000	37	0.4000	48.8000	14.9000	'S 250uL'	'CORDONE ...	'Venoso'
13	2.6000	0.8000	266.4000	132	52.3000	26.8500	44.6000	7.3390	24.9000	46.3000	0.8600	-1.6000	53	37	0	46.4000	15.8000	'S 250uL'	'CORDONE ...	'Venoso'
14	2.6000	0.8000	266.4000	132	52.3000	26.8500	44.6000	7.3390	24.9000	46.3000	0.8600	-1.6000	53	37	0	46.4000	15.8000	'S 250uL'	'CORDONE ...	'Venoso'
15	1.4000	0.5000	282.6000	138	90.1000	26.8500	36.7000	7.4040	48.6000	8	0.8600	-1.5000	91.8000	36	14.7000	41.1000	19.4000	'C 125uL'	'CAPILLARE ...	'Capillare'
16	1.4000	0.5000	282.6000	138	90.1000	26.8500	36.7000	7.4040	48.6000	8	0.8600	-1.5000	91.8000	36	14.7000	41.1000	19.4000	'C 125uL'	'CAPILLARE ...	'Capillare'
17	1.3000	0.5000	277.1000	136	90.6000	26.8500	32	7.4420	51.1000	7.3000	0.8600	-2	92.5000	37	15.3000	38.8000	19.5000	'C 125uL'	'CAPILLARE ...	'Capillare'
18	1.3000	0.5000	277.1000	136	90.6000	26.8500	32	7.4420	51.1000	7.3000	0.8600	-2	92.5000	37	15.3000	38.8000	19.5000	'C 125uL'	'CAPILLARE ...	'Capillare'
19	35	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	'S 250uL'	NaN	'Capillare'
20	35	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	'S 250uL'	NaN	'Capillare'
21	2.4000	0.5000	272.2000	133	85.9000	26.8500	51.4000	7.3490	42.5000	12.8000	0.8600	2.5000	87	37	9	53.1000	17.1000	'C 125uL'	'CAPILLARE ...	'Capillare'
22	2.4000	0.5000	272.2000	133	85.9000	26.8500	51.4000	7.3490	42.5000	12.8000	0.8600	2.5000	87	37	9	53.1000	17.1000	'C 125uL'	'CAPILLARE ...	'Capillare'
23	1.9000	0.6000	275.2000	136	86.4000	26.8500	50	7.3510	46.3000	11.7000	0.8600	1.9000	88.1000	37	7.6000	52.7000	15.7000	'C 125uL'	'CAPILLARE ...	'Non Speci...
24	NaN	0.5000	NaN	NaN	79.8000	26.8500	42.9000	7.4020	35.5000	18.2000	0.8600	NaN	81.4000	37	NaN	NaN	16.1000	'C 125uL'	NaN	'Non Speci...
25	3.5000	0.3000	269.2000	133	78.2000	26.8500	35.2000	7.3820	35.1000	19.8000	0.8600	-3.8000	79.8000	37	7.9000	39.3000	16.7000	'C 125uL'	'CAPILLARE ...	'Non Speci...
26	85	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.4000	NaN	NaN	NaN	25.1000	NaN	'C 125uL'	NaN	'Capillare'

Fig. 9.2 – La struttura completa con 39 parametri, ogni riga è un EGA.

3.3 FUORI RANGE ED ERRORI DI MISURA

Ora che i dati si presentano in un formato su cui si possa lavorare in ambiente Matlab, in questo paragrafo ed in quelli successivi verrà trattata la ricerca di una metodologia che permetta in modo sufficientemente affidabile la discriminazione degli EGA alterati da artefatti da prelievo e quindi non affidabili.

3.3.1 Indagine su possibili correlazioni

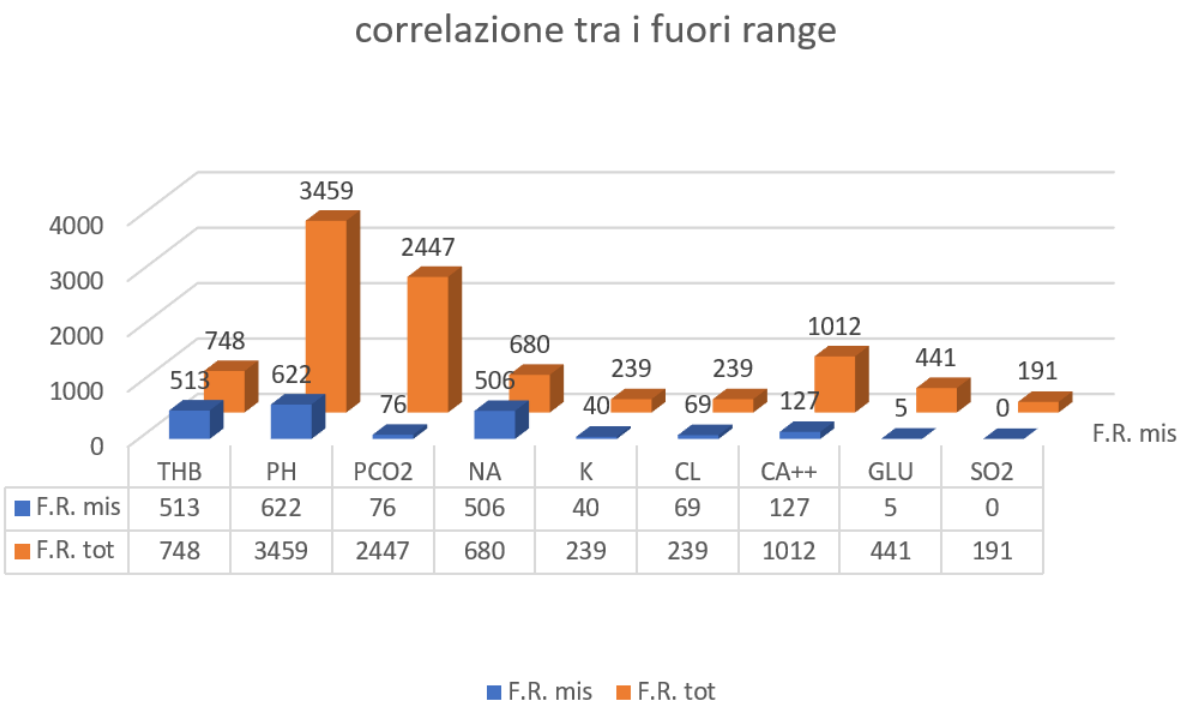
Il primo approccio per l'identificazione automatica degli EGA influenzati da artefatti da prelievo è stato quello di verificare l'entità della correlazione tra parametri misurati e parametri da loro calcolati in modo da individuare una cerchia ristretta di parametri "indicatori" della bontà dell'EGA. Come già accennato in precedenza infatti buona parte dei parametri non viene misurata direttamente dai sensori dell'apparecchio ma calcolata in base ai valori di alcuni parametri direttamente misurati, e se si riuscisse a dimostrare una concreta correlazione si potrebbe individuare un pattern tale da individuare un numero ridotto di parametri misurati su cui basare la discriminazione.

Ho scelto di valutare la correlazione in base alla corrispondenza di valori fuori range tra i parametri misurati e i parametri da loro calcolati, ovvero: se un parametro misurato presenta valori fuori range, saranno fuori range anche i parametri che da esso dipendono?

Il primo step, visibile in Cod. 3 e 4, è stato quello di individuare le posizioni dei fuori range, ovvero in quale riga della struttura dati il parametro in analisi presenta dei valori fuori range. I valori dei range visibili in Cod. 3 e 4 mi sono stati forniti dal personale della TIN, sono valori puramente indicativi, modificabili secondo necessità e avevano il solo scopo di essere dei valori di riferimento su cui testare la correlazione; valori di range diversi porterebbero con molta probabilità a risultati diversi da quelli ottenuti.

Lo script in Cod. 5.1 prende in ingresso la suddetta posizione dei fuori range dei parametri calcolati e l'elenco dei parametri misurati da cui ogni parametro calcolato dipende, in seguito verifica se nella stessa posizione i parametri misurati presentano un valore fuori range. Per fare un esempio: alla riga 309 della struttura completa viene rilevato che il parametro "ABE" ha un valore fuori range, visto che il parametro "ABE" viene calcolato servendosi del valore di THB, PH e PCO2 vengono analizzati i rispettivi valori dei tre parametri nella stessa riga (e quindi stessa EGA) e ne risulta che nessuno dei tre presenta un valore fuori range.

Il tutto viene poi riassunto in una tabella Matlab, Cod. 5.2, in cui per ogni parametro misurato viene indicato il numero delle volte in cui è andato fuori range confrontato col numero delle volte in cui un parametro che da esso dipende è andato fuori range, come si può vedere in Fig. 10.



	THB	PH	PCO2	NA	K	CL	CA++	GLU	SO2
F.R. mis	513	622	76	506	40	69	127	5	0
F.R. tot	748	3459	2447	680	239	239	1012	441	191

Fig. 10 – per F.R. mis si intendono il numero dei valori fuori range di quel parametro quando sono presenti valori fuori range di un altro parametro calcolato in base ad esso, per F.R. tot si intende il numero totale di valori fuori range di tutti i parametri che dipendono dal parametro in questione.

Questo approccio, purtroppo, non ha prodotto risultati conclusivi: non è emerso alcun pattern significativo e nessuna correlazione diretta che potesse evidenziare uno o più parametri "indicatori". Va detto che l'inconclusività dell'approccio può essere ricondotta alla non totale chiarezza delle formule usate dall'apparecchio per calcolare i parametri e/o ai valori scelti per il fuori range dei parametri: di questi ultimi infatti servirebbe una ricerca più approfondita, se non uno studio a parte sulla tematica vista la peculiarità dei soggetti in esame che ricordo essere neonati pretermine stazionati in terapia intensiva neonatale.

Tuttavia, questa elaborazione dei dati non è stata del tutto inutile visto che come prodotto secondario ha fatto emergere statistiche interessanti esposte nel prossimo paragrafo.

3.3.2 Statistiche

NAN	26268
E	7522
(722) - Errore campione	988
(712) - Misurazione FHb non possibile	979
(476) - Misurazione instabile	833
(210) - Errori in calibrazione presenti	824
(374) - Campione inomogeneo nel modulo pH/BG	446
(664) - Problema campione	338
(443) - Ca(7.4) non utilizzabile	296
(581) - OXI spectrum mismatch	248
(719) - Campione insufficiente nel modulo pH/BG	222
(713) - Misurazione ctBil non possibile	128
(721) - Campione insufficiente nel modulo Met II	66
(292) - Torbidità troppo alta	48
(720) - Campione insufficiente nel modulo EI/Met	44
(1010) - Discordanza tra pO2 ed sO2	18
(290) - Attenzione: SHb rilevata	8
(855) - Eccesso di base fuori intervallo	4
(291) - SHb troppo alta	2
(467) - Campione inomogeneo nel modulo EL/Met	2
(468) - Campione inomogeneo nel modulo Met II	2
(677) - Inaffidabile	2
(791) - Il parametro non ha potuto essere misurato	2
(875) - Campione vecchio	2

Fig.11 - Il risultato del ***Cod. 7****: Lista dei messaggi di errore e dei caratteri non numerici rilevati durate la pulizia attuata in* ***Cod. 1.4***
Si nota la prevalenza di dati NaN e dei punti interrogativi, indicati come "E" dopo il processamento in Cod. 1.4

Field	Fuori_range	Perc_range	Nan	Perc_Nan
'B_mmHg'	758	3.2766	63468	73.2870
'CA_2piu'	509	0.7183	15740	18.1751
'CL'	1377	1.7989	10054	11.6094
'COHB'	516	1.3619	48715	56.2516
'GLU'	67	0.0820	4892	5.6488
'HBF'	2397	3.2466	12772	14.7479
'K'	927	1.2662	13388	15.4592
'LAC'	1248	1.5502	6095	7.0379
'METHB'	1197	1.5050	7065	8.1580
'NA'	1781	2.2336	6864	7.9259
'O2HB'	152	0.2800	32323	37.3236
'PCO2'	279	0.3513	7172	8.2816
'PH'	596	0.7548	7643	8.8254
'PO2'	29	0.0364	6870	7.9328
'RHB'	202	0.5329	48697	56.2308
'SO2'	138	0.1729	6802	7.8543
'tBil'	2579	3.3299	9153	10.5690
'THB'	1771	2.1717	5053	5.8347

Field	Fuori_range	Perc_range	Nan	Perc_Nan
'ABE'	167	0.4817	51934	59.9686
'ANION_GAP'	0	NaN	86602	100
'ANION_GAP_K'	360	1.0358	51847	59.8681
'Ca_7_4'	1056	1.3671	9359	10.8069
'cBase_Ecf_ox'	0	0	10246	11.8311
'HCO3'	1951	2.4906	8268	9.5471
'HCO3_st'	0	0	10432	12.0459
'HCT'	428	1.2320	51863	59.8866
'mOsm'	479	1.3705	51651	59.6418
'P50_st'	0	0	60300	69.6289
'SBE'	2	0.0056	51066	58.9663
'tCO2_B'	201	0.5700	51338	59.2804

Fig. 12 – Statistiche numeriche e percentuali dei parametri numerici non immessi. Il parametro con più dati mancanti è l'anion gap col ben 100% di dati NaN, seguito poi dalla pressione atmosferica B con il 73.29%, dal P50(st) col 69.63% e da ABE, ANION GAP K, HCT, mOsm, SBE e tCO2 B che si aggirano sul 60%. Ad avere più fuori range rispetto ai valori scelti è invece il tBil col 3.33% ma anche HBF e B presentano valori simili.

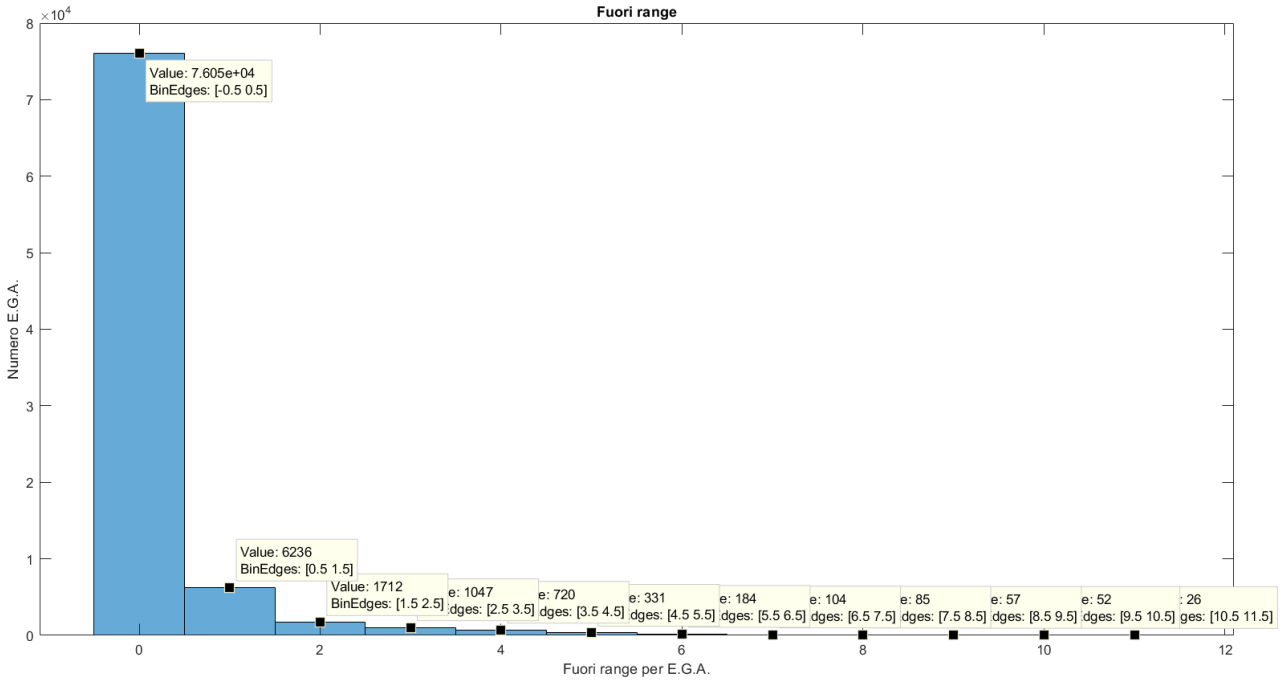


Fig. 13 – Cod. 6 - Numero di parametri con valore fuori range in ogni EGA: la stragrande maggioranza degli EGA non presenta parametri con valori fuori range

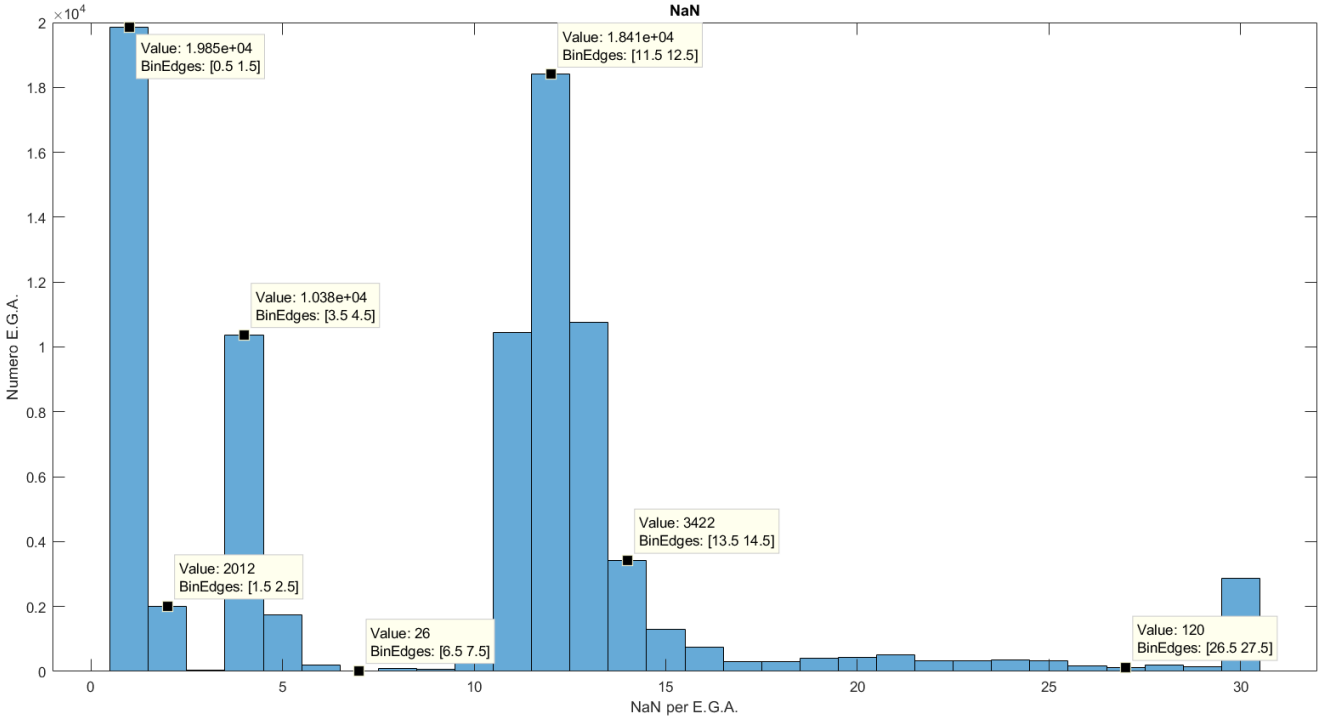


Fig. 14 – Cod. 6 - Numero di parametri con valori mancanti per ogni EGA, si nota una distribuzione distinta: su 86602 EGA, nessun EGA non presentano alcun dato NaN, 19850 ne hanno almeno uno (l'Anion gap), 18410 hanno 12 parametri con valore NaN e due gruppi di circa 11000 EGA presentano 11 e 13 parametri NaN.

3.4 T-TEST

A seguito del non soddisfacente metodo della correlazione tramite il confronto dei valori fuori range, ho cercato di stabilire se la saturazione d'ossigeno costituisse un buon parametro discriminante e per verificare l'ipotesi mi sono servito del t-test.

Il t-test, nello specifico la funzione Matlab `[h,p]=ttest2(x,y)`, è un test statistico che prende in ingresso due set di dati e restituisce una decisione sull'ipotesi nulla che i dati nei due vettori in ingresso x e y provengono da campioni randomici indipendenti da distribuzioni normali con egual "medie" ed eguali ma sconosciute varianze.

- valore *h*: è il risultato del test e può assumere solo due valori, 1 se l'ipotesi nulla è stata rifiutata e 0 se non è stato possibile rifiutare l'ipotesi.
- valore *p*: uno scalare compreso tra 1 e 0, è la probabilità di osservare una statistica del test estrema come il valore osservato sotto l'ipotesi nulla. Piccoli valori di p mettono in dubbio la validità dell'ipotesi nulla. In altre parole, *p* è la probabilità di ottenere risultati estremi come quelli osservati, assumendo che l'ipotesi nulla si a corretta e un basso valore di *p* indica che il risultato è possibile ma non molto probabile sotto l'ipotesi nulla. Un valore di p inferiore a 0,05 indica che i due set di dati sono statisticamente diversi.

Un EGA soggetto ad artefatti da prelievo dovrebbe presentare valori statisticamente diversi da quelli di un EGA genuino, per cui un risultato ottimale sarebbe quello di individuare uno o più valori di sO_2 i cui corrispettivi EGA restituiscano un valore di p inferiore a 0,05 come risultato del t-test.

Come si può vedere in Cod. 8 i due set di dati in ingresso altro non sono altro che i dati EGA separati da un valore soglia di saturazione d'ossigeno, es: un set composto da EGA con saturazione minore o uguale al 70% e l'altro set composto da EGA con saturazione maggiore al 70%. Gli EGA in esame poi sono solo gli EGA bollati come "capillare" nel campo "Tipo" in modo che i valori EGA siano dello stesso tipo di esame. Tramite ciclo for il valore di soglia della saturazione parte da 20% fino ad arrivare a 90% con incrementi del 5%: Nel primo ciclo il ttest2 analizza gli EGA con sO_2 al 20% e gli EGA con sO_2 maggiore al 20%, al secondo ciclo gli EGA con sO_2 minore di 25% e gli EGA con sO_2 maggiore al 25% e così via fino ad arrivare al 90%.

Da ricordare che ogni EGA è composto da 34 parametri numerici più 5 parametri informativi, quindi ogni t-test ad una data soglia di sO_2 è iterato su ognuno dei 34 parametri numerici ottenendo il valore p per ognuno di essi.

Graficando i risultati così ottenuti (Fig. 15) si nota che solo tre valori di sO_2 hanno prodotto il risultato cercato, ovvero un valore *p* inferiore a 0,05 per tutti e 34 i parametri EGA: 55%, 60% e 65%.

Per cercare di confermare e raffinare i risultati ho ripetuto la procedura del t-test cambiando il modo in cui i due set di dati venivano rappresentati dalla sO_2: invece di avere due gruppi caratterizzati da un valore maggiore o minore di una certa sO_2, questa volta i due set di dati erano costituiti da EGA appartenenti ad un intorno di sO_2 , come ad esempio un set costituito dagli EGA con sO_2 compresa tra 20% e 25% e l'altro set costituito dagli EGA con sO_2 compresa tra 25% e 30%. Nel Cod. 9 i due gruppi sono caratterizzati da un intorno di sO_2 del 5% con la sO_2 che parte da 20% e aumenta di volta in volta del 5% fino ad arrivare al 100% mentre nel Cod. 10 l'intorno ha una grandezza pari a una differenza di sO_2 del 10% con la sO_2 che parte da 20% e arriva al 100% aumentando di volta in volta del 10%. Esempio: se nel Cod. 8 alla prima iterazione i due gruppi sono quelli con sO_2 rispettivamente da 20%-25% e 25%-30%, alla seconda iterazione i due gruppi EGA saranno quelli con le saturazioni 25%-30% e 30%-35% e così via fino al 100%, nel Cod. 9 alla prima iterazione i due gruppi sono quelli con la sO_2 rispettivamente al 20%-30% e 30%-40%, poi quelli con 30%-40% e 40%-50% fino al 100%.

Il t-test sull'intorno del 5% (Fig. 17) non ha rilevato discrepanze statistiche, col valore p che rimane sempre molto alto per la maggior parte dei 34 parametri in tutte le iterazioni.
Il t-test sull'intorno del 10% (Fig. 16) presenta una iniziale inflessione del valore p in corrispondenza di una sO_2 del 60% con l'eccezione di 4 parametri per poi confluire con un valore massimo di p di 0,08 in corrispondenza di un intorno del 70% di sO_2.

I risultati ottenuti non sono così decisivi da poter decretare con affidabilità un valore di sO2 per la discriminazione delle EGA, purtuttavia rilevando una discrepanza statistica nelle EGA separate da un valore di sO2 della zona 60-70% in due dei tre test condotti.

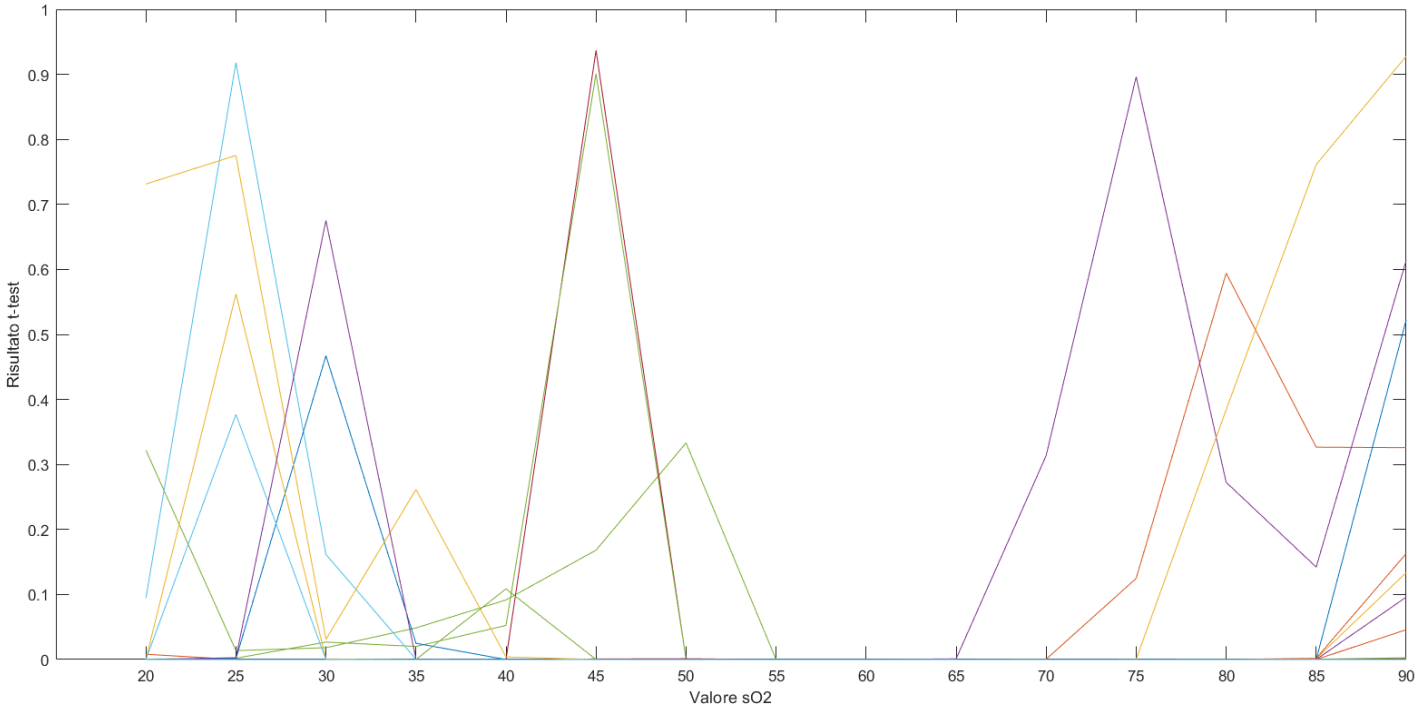


Fig. 15 – Valore di p risultato dal confronto degli EGA divisi in base ad un valore soglia di sO_2, ogni linea colorata rappresenta uno dei 34 parametri. Tutte e 34 le linee scendono a 0 solo a 55%,60% e 65% di sO_2.

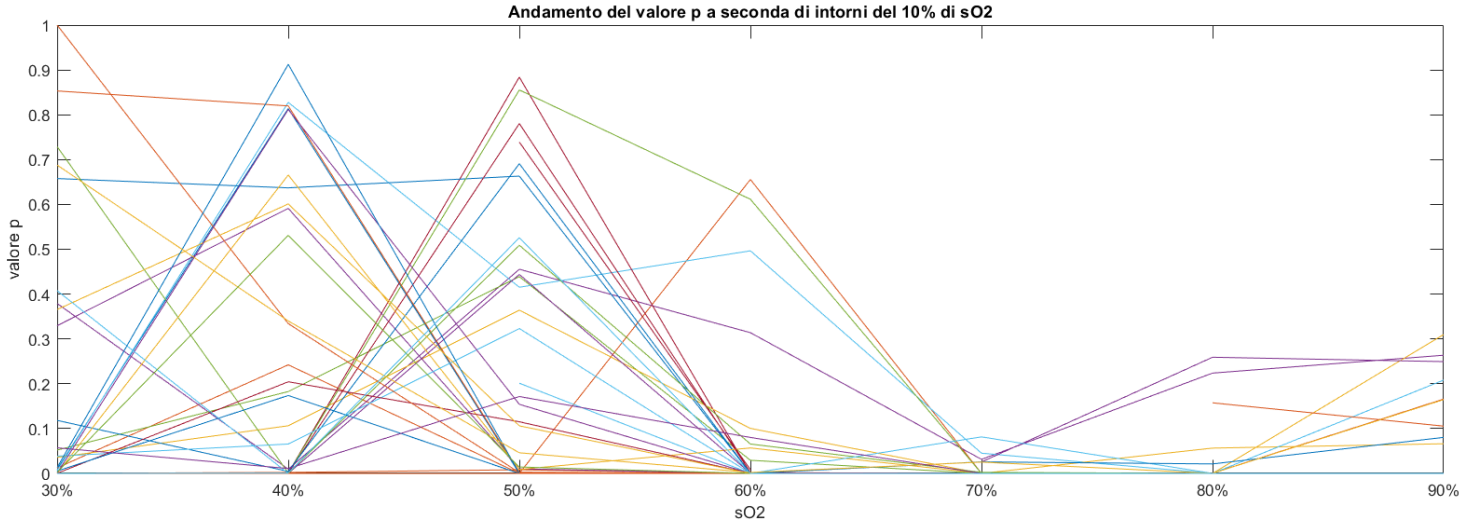


Fig. 16 – Valore di p risultato dal confronto degli EGA divisi in base ad un intorno di sO_2 del 10%, ogni linea colorata rappresenta uno dei 34 parametri.

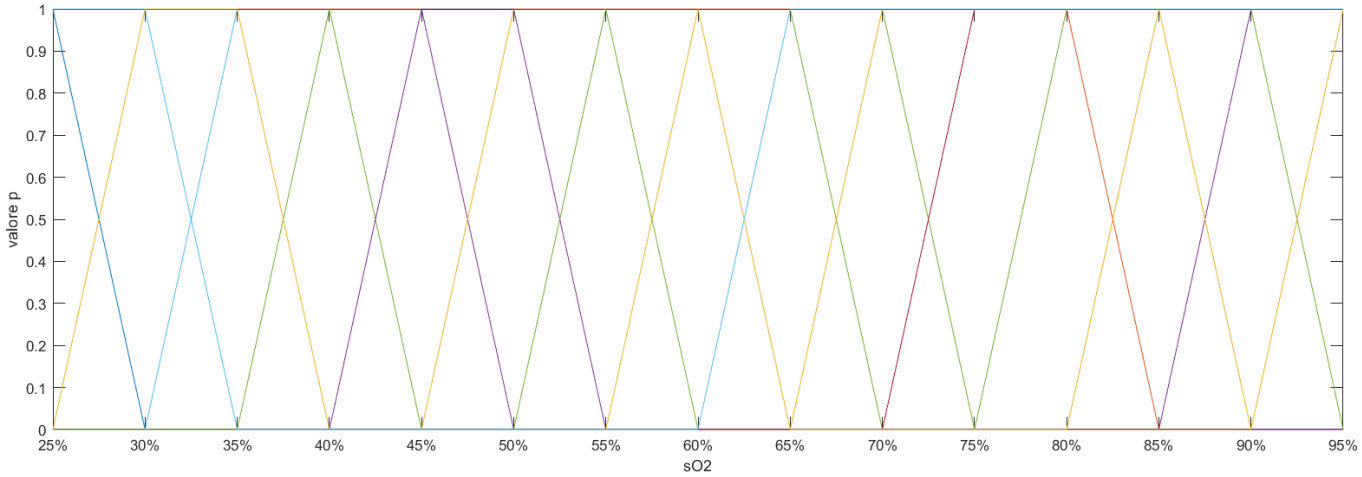


Fig. 17.1 – Valore di p risultato dal confronto degli EGA divisi in base ad un intorno di sO_2 del 5%, ogni linea colorata rappresenta uno dei 34 parametri

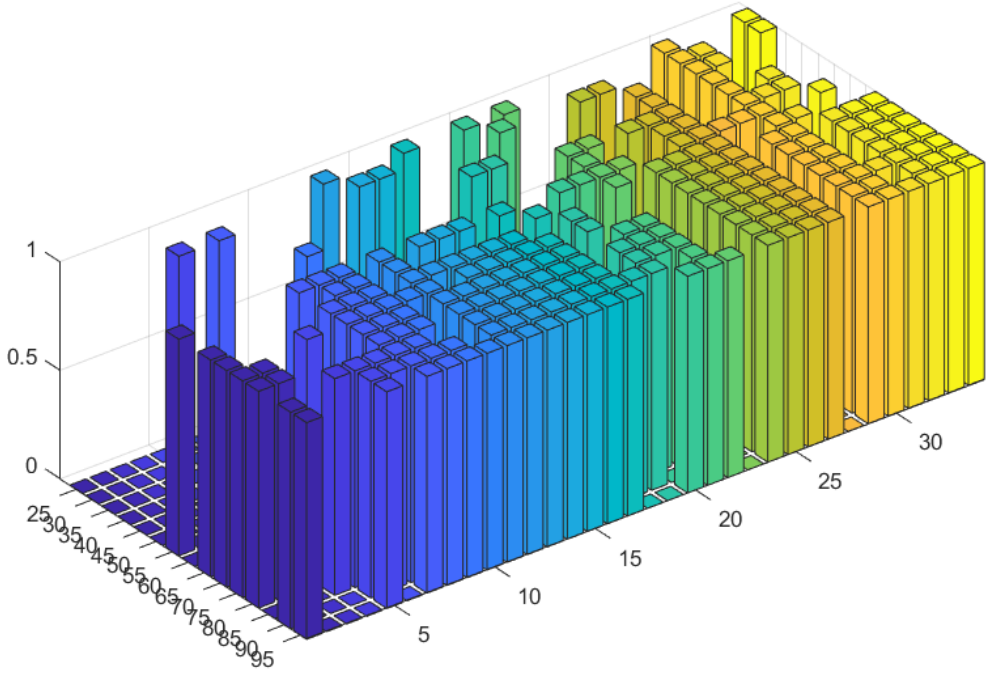


Fig. 17.2 – Visualizzazione in 3D dell'andamento del valore p nel t-test sugli intorni del 5%.

3.5 CLUSTERING TRAMITE K-NEAREST NEIGHBORS

Un ulteriore e definitivo approccio alla ricerca di un metodo affidabile per la discriminazione dei dati EGA è stato l'utilizzo di un classificatore addestrato tramite machine learning.

La scelta del modello di clustering è ricaduta sul modello a K-Nearest Neighbors (knn) poiché è un modello semplice ma che presenta un'ottima accuratezza.

Un classificatore a k-nearest neighbor prende un dato e lo classifica in base al k-nearest neighbor (il k-vicinato più vicino, se k =1 prende il più vicino, k=2 i due più vicini e così via…) iterando il processo più e più volte ed autocorreggendosi in base ai dati forniti in fase di addestramento fino ad arrivare alla creazione di un modello di classificazione più o meno accurato. Tale modello di classificazione può essere usato per classificare nuovi dati dello stesso tipo di quelli forniti in fase di addestramento.

Prima di intraprendere l'addestramento del classificatore è necessaria la creazione di numerosi dati in ingresso (più dati ha per addestrarsi e più il classificatore si addestra bene) con determinate caratteristiche in modo che il classificatore abbia materiale da "studiare". In questo caso visto che lo scopo è quello di individuare EGA non affidabili soggetti ad artefatti da prelievo il materiale di addestramento dovrebbe essere costituito da EGA "buoni" ed EGA "cattivi" ma ciò non è stato possibile visto che non disponevo di materiale simile: un materiale del genere richiederebbe infatti un lavoro a priori di migliaia di EGA analizzate e catalogate da figure mediche competenti secondo i loro standard in modo da creare EGA campioni che vadano a rappresentare EGA normali ed EGA in cui qualcosa è andato storto durante l'analisi.

Vista l'impossibilità di testare il classificatore con lo scopo sopra citato per mancanza di materiale adatto, ho testato le capacità di tale metodo di classificazione su dati EGA addestrandolo per un altro scopo: la suddivisione di EGA di "Tipo" capillare in venoso e arterioso.

Per addestrare il classificatore ho creato due set di dati in formato *table* (tabella)(Cod. 12.1): gli EGA venosi costituiti dai EGA con "Tipo" venoso e gli EGA arteriosi costituiti da tutti gli EGA con "Tipo" arterioso, cordone ombelicale e scalpo fetale (su indicazione del personale della TIN), da entrambe i set ho rimosso quelli che presentavamo come "PRELIEVO" urine(Cod. 11). In entrambe i set ho aggiunto un nuovo campo chiamato "Tic" per contrassegnare il tipo di dato e segnalarlo al modello in training tramite la specifica *'ClassNames'* (Cod. 12.1) con 0 che corrisponde ai venosi e 1 agli arteriosi. Ho poi unito i due set EGA arteriosi e venosi in un'unica tabella, con ordine randomizzato ed eliminando qualsiasi riga presentasse almeno un NaN in modo da avere dati pieni.

Come si può vedere in Cod. il classificatore è stato addestrato con la funzione *fitcknn* con varie ottimizzazioni:

- Distance= Euclidean
- NumNeighbors= 1
- DistanceWeight= SquaredInverse
- Standardize= true (1)
- ClassNames= [0 ; 1]

Si noti la presenza dei valori identificativi di riferimento (*ClassNames*) 0 e 1 ad indicare venoso e arterioso, l'impostazione alla ricerca del solo vicinato più vicino (*NumNeighbors=1*) e la pesatura della distanza del vicinato all'inverso del quadrato (*SquaredInverse*).

Ho poi testato (Cod. 12.2) i modelli classificatori, ottenuti variando di volta in volta i parametri di ottimizzazione appena menzionati, utilizzando la funzione *"predict"* sugli EGA capillari e gli EGA capillari che non contenessero alcun NaN.

1. `DistanceWeight=SquaredInverse, NumNeighbors=1,` NaN presenti nella tabella EGA capillare:

- EGA capillari totali in ingresso:……**71842**
- EGA capillari classificati come **arteriosi**:……..**3356** (media sO_2=**86,03%**)
- EGA capillari classificati come **venosi**:……..**68486** (media sO_2=**81,84%)**

2. `DistanceWeight=SquaredInverse, NumNeighbors=1,` tabella capillare EGA senza alcun NaN:

- EGA capillari totali in ingresso:…… **15124**
- EGA capillari classificati come **arteriosi**:……**3356** (media sO_2=**86,03%**)
- EGA capillari classificati come **venosi**:……...**11768** (media sO_2=**82,42%)**

3. `DistanceWeight=Equal, NumNeighbors=1,` NaN presenti nella tabella EGA capillare:

- EGA capillari totali in ingresso:……**71842**
- EGA capillari classificati come **arteriosi**:……**3356** (media sO_2=**86,04%**)
- EGA capillari classificati come **venosi**:……...**68486** (media sO_2=**81,84%)**

4. `DistanceWeight=Equal, NumNeighbors=1,` tabella capillare EGA senza alcun NaN:

- EGA capillari totali in ingresso:……**15124**
- EGA capillari classificati come **arteriosi**:……**3356** (media sO_2=**86,04%**)
- EGA capillari classificati come **venosi**:……...**11768** (media sO_2=**82,42%**)

5. `DistanceWeight=SquaredInverse, NumNeighbors=5,` NaN presenti nella tabella EGA capillare:

- EGA capillari totali in ingresso:……**71842**
- EGA capillari classificati come **arteriosi**:……**2898** (media sO_2=**86,51%**)
- EGA capillari classificati come **venosi**:…….. **68944** (media sO_2=**81,84%**)

6. `DistanceWeight=SquaredInverse, NumNeighbors=5,` tabella capillare EGA senza alcun NaN:

- EGA capillari totali in ingresso:……**15124**
- EGA capillari classificati come **arteriosi**:……**2898** (media sO_2=**86,51%**)
- EGA capillari classificati come **venosi**:……...**12226** (media sO_2=**82,45%**)

Esaminando l'accuratezza dei modelli tramite l'uso di *kfoldLoss* e *crossval* (Cod. 12.2), il modello di clustering con le caratteristiche del punto 2 è risultato essere quello con l'accuratezza migliore, pari al 98,35%: L'assenza dei NaN comporta infatti la presenza di più dati su cui basare il riconoscimento e l'impostazione ad un solo Neighbor restringe al minimo l'incertezza durante il training.

Da evidenziare è il fatto che per tutti i modelli di classificazione la presenza o assenza dei NaN non influisce sul numero di EGA riconosciuti come arteriosi ma solo su quelli venosi, indicando quindi che i prelievi EGA problematici, e quindi soggetti ad artefatti da prelievo, sono riconducibili ai capillari venosi.

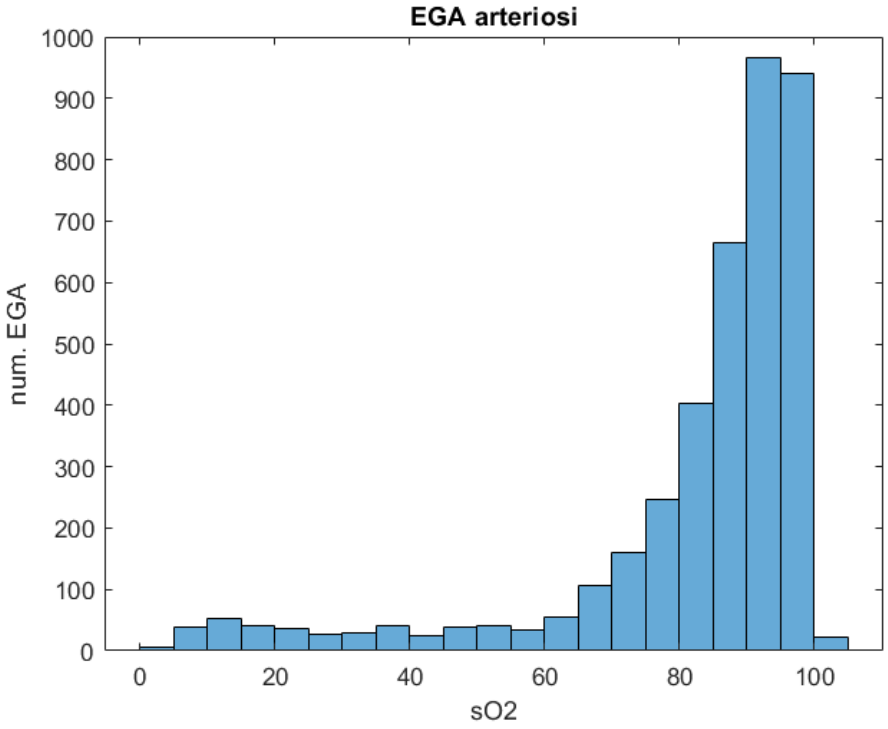


Fig. 18 – Distribuzione della saturazione di ossigeno negli EGA arteriosi, , valore medio sO_2=82,69%

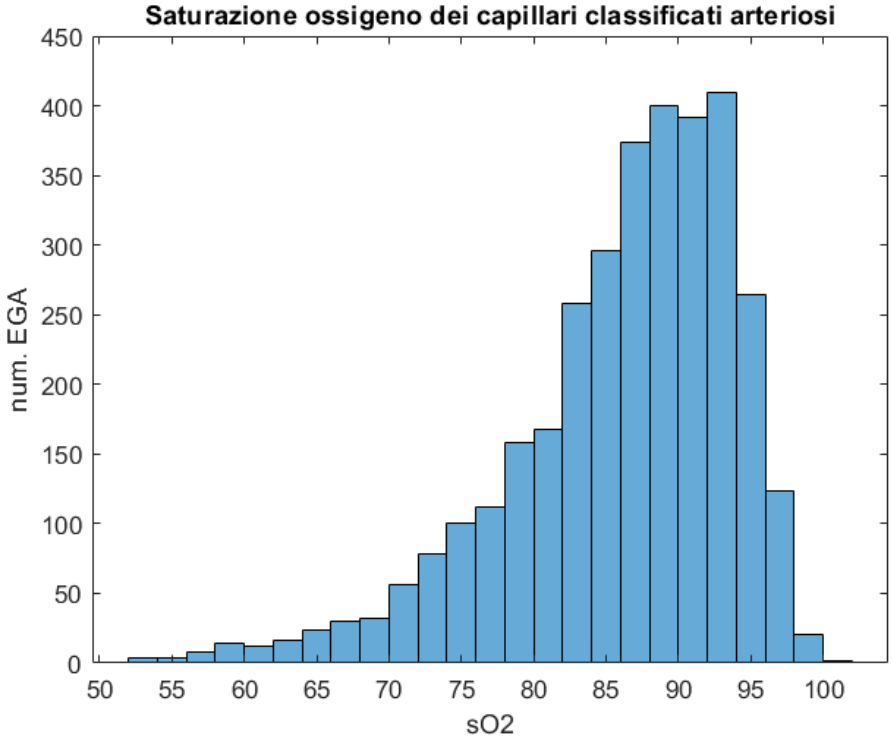


Fig. 19 - Distribuzione della saturazione di ossigeno negli EGA capillari classificati come arteriosi, valore medio sO_2=86,03%

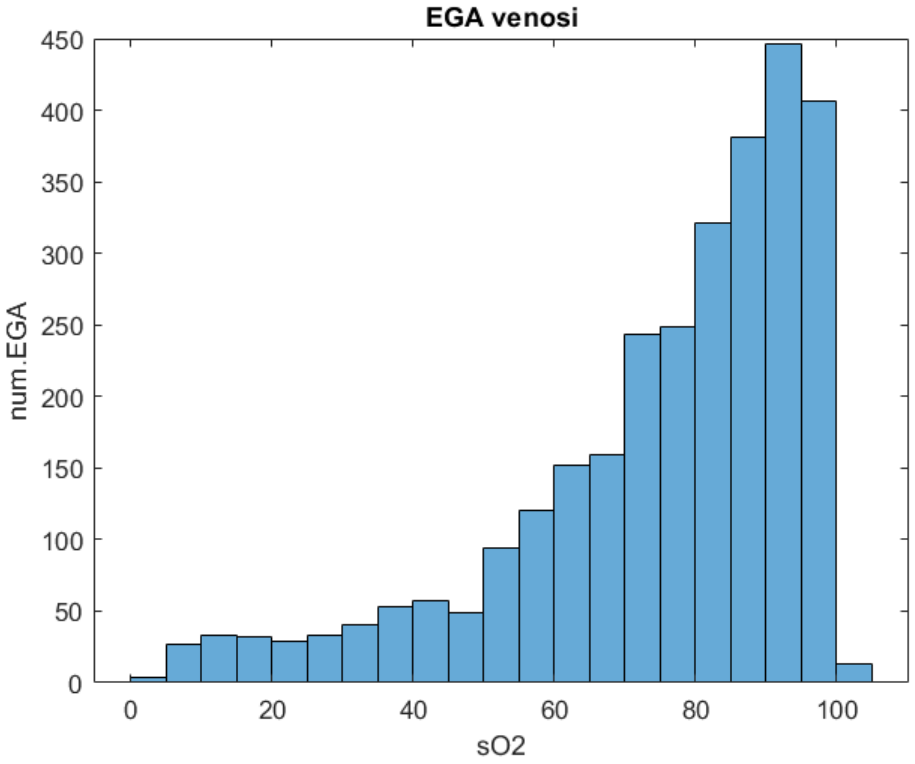


Fig. 20 – Distribuzione della saturazione di ossigeno negli EGA venosi, valore medio sO_2=75,38%

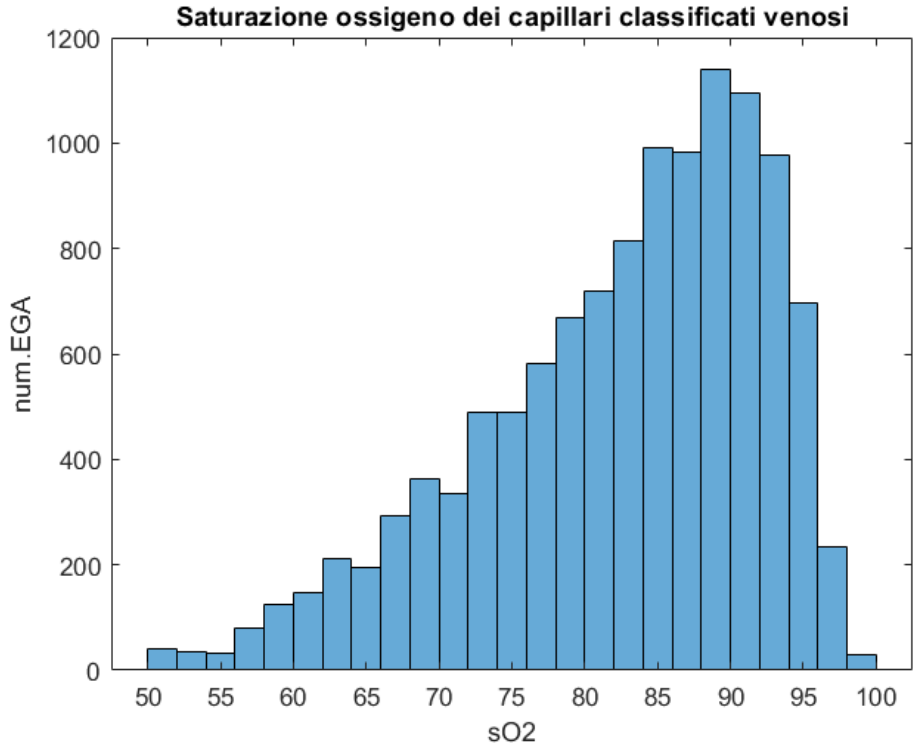


Fig. 21 - Distribuzione della saturazione di ossigeno negli EGA capillari classificati come venosi, valore medio sO_2=82,42%

4 CONCLUSIONI E SVILUPPI FUTURI

Lo scopo principale di questo lavoro era quello di creare un algoritmo in ambiente Matlab per l'elaborazione dei dati risultati dalle emogasanalisi effettuate sui nati pretermine stazionati in Terapia Intensiva Neonatale e la ricerca di un metodo affidabile per la discriminazione automatizzata delle EGA in base a parametri arbitrari dettate dalle esigenze del personale medico.

Per la prima parte di elaborazione e gestione dei dati, l'uso di una struttura Matlab si è dimostrato di grande efficacia sia per la possibilità di inglobare dati di tipo diverso sia per la capacità di gestire una tale mole di dati. La creazione e l'immissione dei dati EGA in questa struttura ha anche permesso l'unificazione dei dati in un unico formato comune: se prima infatti i dati EGA si presentavano in formato diverso in base al periodo temporale in cui erano state effettuate le analisi, ora è possibile consultare e lavorare sui dati in un formato pulito, unico e standardizzato a 39 parametri, una strutturazione che rende più semplici ed immediate le varie operazioni che coinvolgono tali dati. Un esempio è la possibilità di rappresentare graficamente e in vari stili l'andamento dei valori di uno o più parametri delle EGA di un particolare paziente, periodo o interesse. La struttura è anche facilmente intercambiabile in formato tabellare (*table*) a seconda delle esigenze per una rappresentazione più semplificata ma più veloce sotto il punto di vista computazionale: durante il lavoro svolto infatti è stato evidente che quando venivano eseguite operazioni su tutte le EGA, queste venivano svolte molto più velocemente se i dati erano nel formato *table.*

Voglio sottolineare poi che il tutto il lavoro svolto per la pulizia e la trasposizione dei dati EGA originali in questo formato *struttura* Matlab costituisce infine una solida base di partenza per qualsiasi lavoro futuro si possa desiderare svolgere su questi dati EGA.

Per quanto riguarda la ricerca di un metodo affidabile di discriminazione sono stati tentati tre approcci, ognuno dei quali ha portato i propri risultati.

La ricerca di una correlazione tra i parametri misurati e quelli calcolati non ha prodotto risultati chiari o decisivi, non escludendo che possa rivelarsi un metodo valido a seguito di un approccio diverso. La mancanza di risultati conclusivi di questo approccio potrebbe essere dovuta ai valori dei limiti di range scelti o alla non chiarezza di alcune delle equazioni con cui l'apparecchio calcola alcuni parametri. L'elaborazione ha comunque fatto emergere statistiche interessanti (Fig 11-14) come il fatto che non sia presente alcun valore del parametro "Anion Gap" e manchino oltre il 70% delle rilevazioni della pressione atmosferica, il fatto che la maggior parte delle EGA non presenta parametri con valori fuori range ed il fatto che un numero significativo di EGA presenta 11, 12 o 13 parametri senza valore.

L'indagine tramite t-test ha indicato una possibile differenza statistica nelle EGA separate da un valore di saturazione intorno al 60/70% ma l'approccio scelto non ha permesso di ottenere risultati conclusivi: le tre tipologie di t-test effettuati hanno infatti dato risultati diversi o a risultati che comunque sia non hanno individuato in maniera certa un valore discriminante di sO_2.

La creazione di un modello classificatore addestrato in machine learning con modello a K-Nearest Neighbors si è dimostrato di gran lunga il metodo più affidabile e funzionale per la discriminazione di questa mole di dati di questo tipo. Questo modello di classificazione è stato testato con la distinzione delle EGA "*capillare*" in "*venoso*" e "*arterioso*", compito che ha svolto con un'accuratezza del 98%.

Inoltre, questa classificazione ha fatto emergere il dato forse più importante di tutto l'elaborato: Le EGA che presentano parametri con valori NaN sono state classificate tutte come venose. Questo fatto sembrerebbe indicare che tutte le EGA problematiche, e quindi probabilmente soggette ad artefatto da prelievo, sono riconducibili solo ed esclusivamente alle EGA di tipo venoso.

Partendo da queste fondamenta, si potranno sviluppare studi più approfonditi sui dati EGA: l'approccio alla ricerca della correlazione e del t-test possono essere rivisitati a fronte dei dati e risultati qui raccolti mentre il machine learning, a seguito degli ottimi risultati ottenuti, può essere utilizzato con affidabilità su altre ricerche o compiti. In particolare, il machine learning potrebbe essere impiegato nella discriminazione delle EGA soggette ad artefatto da prelievo, con la necessità di uno studio dedicato per l'individuazione di EGA adatte al training del modello.

5 APPENDICE CODICI MATLAB

```
function [x,ERRORE]=struttura(filename,y) %la y posta a 0 se non si deve creare una struttura
generale

[ega ega ega]  = xlsread(filename);   %apro file
tit=ega(1,:);                         %estrapolo i nomi dei parametri(titoli) nel file
ega=ega(2:end,:);                     %tolgo i titoli (ph, pco2...etc)
```

Cod. 1.1- Sintassi della funzione e acquisizione dell'input

```
nomi={'Codice' 'Data Analisi' 'ABE mmol/L' 'cBase(B) mmol/L'...
     'Anion gap (K+) mmol/L' 'GAP (K+) '  'ANION GAP mmol/L'...
     'B mmHg' 'Ca(7.4) mg/dL' 'CA++ mg/dL' 'Ca2+ mg/dL'...
     'cBase(Ecf,ox) mmol/l' 'CL- mmol/L' 'COHB %' 'CREA mg/dl'...
     'FIO2 %' 'GLU mg/dL' 'HBF %' 'HCO3- (st) mmo/L'...
     'HCO3-(st) mmo/L' 'SBC mmol/L' 'HCO3- mmo/L' 'HCT %'...
     'K+ mmol/L' 'LAC mmol/L' 'METHB %' 'mOsm mmol/Kg'...
     'NA+ mmol/L' 'O2HB ' 'O2HB %' 'P50(st) mmHg' 'PCO2 mmHg'...
     'PH ' 'PO2 mmHg' 'RHB %' 'RQ  ' 'SBE mmol/L'...
     'SO2 %' 'T ° Cel' 'tBil mg/dL' 'tCO2(B) Vol%' ...
     'THB g/dL' 'Mod. di misura  ' 'PRELIEVO ' 'Tipo' }; %sono tutti i parametri inclusi i doppioni


  titexp={'Codice' 'Data' 'ABE' 'Doppione_1'...
      'ANION_GAP_K' 'Doppione_2' 'ANION_GAP' 'B_mmHg'...
      'Ca_7_4' 'CA_2piu' 'Doppione_3'...
      'cBase_Ecf_ox' 'CL' 'COHB'...
      'CREA' 'FIO2' 'GLU' 'HBF'...
      'HCO3_st' 'Doppione_4' 'Doppione_5' 'HCO3' 'HCT' 'K' 'LAC'...
      'METHB' 'mOsm' 'NA' 'O2HB' 'Doppione_6'...
      'P50_st' 'PCO2' 'PH' 'PO2'...
      'RHB' 'RQ' 'SBE' 'SO2' 'T_Cel' 'tBil'...
      'tCO2_B' 'THB' 'Mod_di_misura' 'PRELIEVO' 'Tipo'};   %nome dei campi della struttura estesa
```

Cod. 1.2- Nomi di tutti i parametri presenti in tutti i file Excel e nomi dei campi della struttura

```
x=struct();  %inizializzo la struttura vuota
for c=1:length(nomi)
    inde=(strfind(tit,nomi(1,c))); %in quale colonna del file excel trovo il c-esimo parametro
contenuto nell'array "nomi"
    a=zeros(size(inde));
    for m=1:size(inde,2)
        if cell2mat(inde(m))==1
            a(m)=1;
        end
    end
    b=find(a>0) ;
    if b>0
        index(c,1)=b; %se ha trovato riscontro, index contiene ora il numero della colonna del file
Excel che contiene il parametro cercato

    else
        index(c,1)=0; %se non ha trovato riscontro index contiene 0
    end
    for m=1:size(ega,1)
        if index(c,1)>0
            x(m).(char(titexp(c)))=cell2mat(ega(m,index(c,1))); %inserisco nel campo della
struttura il corrispettivo valore della colonna excel
        else
            x(m).(char(titexp(c)))=NaN;  %se non trova corrispondenze crea lo stesso il campo,
riempiendolo di NaN
        end

    end
end
```

Cod. 1.3- Creazione della struttura e inserimento dei dati

```
fld=fieldnames(x);
ERRORE=struct();
for c=3:length(fld)-3
    for r=1:length(x)
        a=x(r).(char(fld(c)));
        if not(isnan(a))
            x(r).(char(fld(c)))=strrep(a,'?','');
        end
    end
end % sostituisco '' ai '?'
for c=3:length(fld)-3
    for r=1:length(x)
        a=x(r).(char(fld(c)));
        if not(isnan(a))
            x(r).(char(fld(c)))=strrep(a,'.....','E');
        end
    end
end % sostituisco 'E' ai '.....'
for c=3:length(fld)-3
    for r=1:length(x)
        a=x(r).(char(fld(c)));
        if not(isnan(a))
            x(r).(char(fld(c)))=strrep(a,',','.');
```

```
        end
    end
end % sostituisco i punti alle virgole
z=1;
for r=1:length(x)
   for c=3:length(fld)-3
        a=x(r).(char(fld(c)));
        v=isstrprop(a,'alpha');    %crea un vettore contenente 1 in corrispondenza delle lettere
        l=sum(v);
        if l>0
            x(r).(char(fld(c)))=NaN;
            ERRORE(z).riga=(r+y);
            ERRORE(z).field=fld(c);
            ERRORE(z).errore=a;
            z=z+1;
        end
    end
end %sostituisco NaN alle scritte di errore escludendo codice,data e gli ultimi 3 campi. creo i
file errore
for c=1:length(fld)
    for r=1:length(x)
        a=x(r).(char(fld(c)));
        if isempty(a)    %se è vuoto
           x(r).(char(fld(c)))=NaN;
        end
    end
end %sostituisco NaN ai campi vuoti
for c=3:length(fld)-3
    for r=1:length(x)
        if not(isnan(x(r).(char(fld(c)))))
           x(r).(char(fld(c)))=str2num(x(r).(char(fld(c))));
        end
    end
end % converto da stringhe a numero per plottare
```

Cod. 1.4- Pulizia e formattazione dei dati

```
seldop={'ABE' 'Doppione_1' NaN;...
    'ANION_GAP_K' 'Doppione_2' NaN;...
    'CA_2piu' 'Doppione_3' NaN;...
    'HCO3_st' 'Doppione_4' 'Doppione_5';...
    'O2HB' 'Doppione_6' NaN};

for r=1:5
    for d=2:3
        if not(isnan(cell2mat(seldop(r,d))))
            a=[x.(char(seldop(r,d)))]; %per ottenere i valori del campo su un vettore
            if min(isnan(a))<1 %ovvero se non è tutto NaN (e quindi è quello da prendere)
                for g=1:size(ega,1)
                    x(g).(char(seldop(r,1)))=x(g).(char(seldop(r,d)));
                end
                x=rmfield(x,char(seldop(r,d)));
            else
                x=rmfield(x,char(seldop(r,d))); %per rimuovere un campo
            end
            clear a;
```

```
        end
    end
end %se un doppione contiene valori,li trasferisco ed elimino il campo, altrimenti elimino il campo
```

Cod. 1.5- Eliminazione dei parametri doppione dalla struttura

```
addpath('emogas')
a= dir('emogas'); %struttura che contiele i nomi dei file xls nella cartella emogas
c=a(3).name;  %i nomi dei file partono dalla terza riga
[x,err]=struttura(c,0); %inizializzo la struttura col primo file. la seconda variabile di input
rappresenta la lunghezza da aggiungere alle righe delle strutture errore in modo da far riferimento
alla struttura generale
d = fieldnames(x);
for e=4:length(a)
    b=a(e).name;
    [s,err1]=struttura(b,length(x));
    x=[x,s];
    err=[err,err1];
end
```

Cod. 2-Unione delle varie strutture dati e strutture errore usando la funzione "struttura" precedentemente creata

```
function [compare,control,nan,israng e]=statmis(x)
israng e=[];
nan=[];
control=struct;
mis= {'B_mmHg' 'CA_2piu' 'CL' 'COHB' 'GLU'...
    'HBF' 'K' 'LAC' 'METHB' 'NA' 'O2HB' 'PCO2'...
    'PH' 'PO2' 'RHB' 'SO2' 'tBil' 'THB'}; %i parametri direttamente misurati

range=[600.00,800.00;...B_mmhg
    0.10,10.00;...Ca2+ mg/dl
    60.00,160.00;...CL mmol/l
    0.10,10.00;...COHB %
    1.00,600.00;...GLU mg/dl
    0.10,100.00;...HBF %
    1.50,12.00;...K mmol/l
    0.10,10.00;...lac mmol/l
    0.10,5.00;...methb %
    110.00,160.00;...Na mmol/l
    0.10,100.00;...O2HB %
    0.10,100.00;...pCO2 mmHg
    6.75,7.60;...PH
    0.10,200.00;...pO2 mmHg
    0.10,80.00;...RHB %
    0.10,100.00;...sO2%
    0.10,60.00;...tBil mg/dl
    4.00,28.00;...THB g/dl
    ];

for e=1:length(mis)
    l=1;
   for i=1:length(x)
```

```
        a=x(i).(char(mis(e)));
        b=range(e,1);
        c=range(e,2);
        if or(a<b,a>c)
            isrange(i,e)=1;
            control(l).(char(mis(e)))=i;
            l=l+1;
        else
            isrange(i,e)=0;
        end
        if isnan(a)
            nan(i,e)=1;
        else
            nan(i,e)=0;
        end
    end
end  % creazione matrici con valori 1 corrispondenti alla presenza di valori fuori range e nan.
for z=1:length(mis)
    overrange(1,z)=sum(isrange(:,z));
    totnan(1,z)=sum(nan(:,z));
end %vettori contenenti il numero totale di fuori range e nan.
compare=struct;
for y=1:length(mis)
    compare(y).Field=mis(y);
    compare(y).Fuori_range=overrange(y);
    w=((overrange(y)/(length(x)-totnan(y)))*100);
    compare(y).Perc_range=w;
    compare(y).Nan=totnan(y);
    g=(totnan(y)/length(x))*100;
    compare(y).Perc_Nan=g;
end %struttura contenente i risultati.
```

Cod. 3 – Lo script statmis prende in ingresso una struttura dati mentre in uscita "control" è una struttura contenente la posizione(righe) dei parametri fuori range e "compare" è una struttura che per ogni parametro misurato indica il numero di fuori range e numero di Nan con relative percentuali.

```
function [compare,control,nan,isrange]=statcalc(x)

calc={'ABE' 'ANION_GAP' 'ANION_GAP_K'...
    'Ca_7_4' 'cBase_Ecf_ox' 'HCO3' 'HCO3_st'...
    'HCT' 'mOsm' 'P50_st' 'SBE' 'tCO2_B'}; %elenco dei parametri che non sono direttamente misurati

range=[-25.00,25.00;...ABE mmol/l
       -999.99,999.99;...anion gap mmol/l
       -20.00,30.00;...gap k+ mmol/l
       2.00,8.00;...Ca7.4 mg/dl
       -999.99,999.99;...cbase(ecf ox)mmol/l
       10.00,50.00;...hco3 mmol/l
       -999.99,999.99;...hco3 st mmol/l
       10.00,80.00;...hct %
       200.00,400.00;...mosm mmol/Kg
       -999.99,999.99;...p50 st mmHg
       -999.99,999.99;...sbe mmmol/l
       0.10,100.00;...tco2(b)%
       ];
```

```
isrange=[];
nan=[];
control=struct;

   for e=1:length(calc)
    l=1;
   for i=1:length(x)
       a=x(i).(char(calc(e)));
       b=range(e,1);
       c=range(e,2);
       if or(a<b,a>c)
           isrange(i,e)=1;
           control(l).(char(calc(e)))=i;
           l=l+1;
       else
           isrange(i,e)=0;
       end
       if isnan(a)
           nan(i,e)=1;
       else
           nan(i,e)=0;
       end
   end
end
   for z=1:length(calc)
    overrange(1,z)=sum(isrange(:,z));
    totnan(1,z)=sum(nan(:,z));
end %vettori contenenti il numero totale di fuori range e NaN
compare=struct;
for y=1:length(calc)
    compare(y).Field=calc(y);
    compare(y).Fuori_range=overrange(y);
    w=((overrange(y)/(length(x)-totnan(y)))*100);
    compare(y).Perc_range=w;
    compare(y).Nan=totnan(y);
    g=(totnan(y)/length(x))*100;
    compare(y).Perc_Nan=g;
end %struttura contenente i risultati.
```

Cod. 4 - – Lo script statcalc prende in ingresso una struttura dati mentre in uscita "control" è una struttura contenente la posizione(righe) dei parametri fuori range e "compare" è una struttura che per ogni parametro calcolato indica il numero di fuori range e numero di Nan con relative percentuali.

```
load('posiz_fuorirange_misurati.mat');
load('posiz_fuorirange_calcolati.mat');
%questi parametri calcolati dipendono da:
parABE={'THB' 'PH' 'PCO2'};
parGAPK={'NA' 'K' 'CL' 'PCO2' 'PH'};
parCA74={'CA_2piu' 'PH'};
parHCO3={'PCO2' 'PH'};
parHCT={'THB'};
parMOSM={'NA' 'GLU'};
parTCO2B={'PCO2' 'THB' 'PH' 'SO2'};
parANIONGAP={'NA' 'CL' 'PCO2' 'PH'};
parCBASEECF={'THB' 'SO2' 'THB' 'PH' 'PCO2'};
parHCO3ST={'SO2' 'THB' 'PH' 'PCO2'};
```

```
parSBE={'THB' 'PH' 'PCO2'};


a=[contr_calc.ABE];
ABE=zeros(length(a),length(parABE));
for x=1:length(a)
    y=contr_calc(x).ABE;
    for i=1:length(parABE)
        b=find([contr_mis.(char(parABE(i)))] == y);
        if not(isnan(b))
            ABE(x,i)=1;
        else
            ABE(x,i)=0;
        end
    end
end
clear a
ABE = array2table(ABE,'VariableNames',parABE);

a=[contr_calc.ANION_GAP_K];
GAPK=zeros(length(a),length(parGAPK));
for x=1:length(a)
    y=contr_calc(x).ANION_GAP_K;
    for i=1:length(parGAPK)
        b=find([contr_mis.(char(parGAPK(i)))] == y);
        if not(isnan(b))
            GAPK(x,i)=1;
        else
            GAPK(x,i)=0;
        end
    end
end
clear a
GAPK = array2table(GAPK,'VariableNames',parGAPK);

a=[contr_calc.Ca_7_4];
CA74=zeros(length(a),length(parCA74));
for x=1:length(a)
    y=contr_calc(x).Ca_7_4;
    for i=1:length(parCA74)
        b=find([contr_mis.(char(parCA74(i)))] == y);
        if not(isnan(b))
            CA74(x,i)=1;
        else
            CA74(x,i)=0;
        end
    end
end
clear a
CA74 = array2table(CA74,'VariableNames',parCA74);

a=[contr_calc.HCO3];
HCO3=zeros(length(a),length(parHCO3));
for x=1:length(a)
    y=contr_calc(x).HCO3;
    for i=1:length(parHCO3)
        b=find([contr_mis.(char(parHCO3(i)))] == y);
        if not(isnan(b))
            HCO3(x,i)=1;
```

```
        else
            HCO3(x,i)=0;
        end
    end
end
clear a
HCO3 = array2table(HCO3,'VariableNames',parHCO3);

a=[contr_calc.HCT];
HCT=zeros(length(a),length(parHCT));
for x=1:length(a)
    y=contr_calc(x).HCT;
    for i=1:length(parHCT)
        b=find([contr_mis.(char(parHCT(i)))] == y);
        if not(isnan(b))
            HCT(x,i)=1;
        else
            HCT(x,i)=0;
        end
    end
end
clear a
HCT = array2table(HCT,'VariableNames',parHCT);

a=[contr_calc.mOsm];
MOSM=zeros(length(a),length(parMOSM));
for x=1:length(a)
    y=contr_calc(x).mOsm;
    for i=1:length(parMOSM)
        b=find([contr_mis.(char(parMOSM(i)))] == y);
        if not(isnan(b))
            MOSM(x,i)=1;
        else
            MOSM(x,i)=0;
        end
    end
end
clear a
MOSM = array2table(MOSM,'VariableNames',parMOSM);

a=[contr_calc.tCO2_B];
TCO2B=zeros(length(a),length(parTCO2B));
for x=1:length(a)
    y=contr_calc(x).tCO2_B;
    for i=1:length(parTCO2B)
        b=find([contr_mis.(char(parTCO2B(i)))] == y);
        if not(isnan(b))
            TCO2B(x,i)=1;
        else
            TCO2B(x,i)=0;
        end
    end
end
clear a
TCO2B = array2table(TCO2B,'VariableNames',parTCO2B);
```

Cod. 5.1 – Ricerca della correlazione tra i valori di fuori range dei parametri misurati e calcolati.

```
righe= {'THB';'PH';'PCO2';'NA';'K';'CL';'CA++';'GLU';'SO2'};
thb1=sum(ABE{:,1})+sum(HCT{:,1})+sum(TCO2B{:,2});
thb2=height(ABE)+height(HCT)+height(TCO2B);
ph1=sum(ABE{:,2})+sum(GAPK{:,5})+sum(CA74{:,2})+sum(HCO3{:,2})+sum(TCO2B{:,3});
ph2=height(ABE)+height(GAPK)+height(CA74)+height(HCO3)+height(TCO2B);
pco1=sum(ABE{:,3})+sum(GAPK{:,4})+sum(HCO3{:,1})+sum(TCO2B{:,1});
pco2=height(ABE)+height(GAPK)+height(HCO3)+height(TCO2B);
na1=sum(GAPK{:,1})+sum(MOSM{:,1});
na2=height(GAPK)+height(MOSM);
k1=sum(GAPK{:,2});
k2=height(GAPK);
cl1=sum(GAPK{:,3});
cl2=height(GAPK);
ca1=sum(CA74{:,1});
ca2=height(CA74);
glu1=sum(MOSM{:,2});
glu2=height(MOSM);
so1=sum(TCO2B{:,4});
so2=height(TCO2B);
s=[thb1; ph1;pco1;na1;k1;cl1;ca1;glu1;so1];%è il numero dei fuori range misurati che sono presenti
quando è fuori range un calcolato
h=[thb2;ph2;pco2;na2;k2;cl2;ca2;glu2;so2];%fuori range totali che dipendono da quella variabile
misurata

%  Esempio:tramite il Na vengono calcolate 2 parametri(gapK+ e mOsm),
%  il gapK+ ha 239 fuori range ed il mOsm 441 per un totale di 680. di
%  questi 680 fuori range solo 506 hanno anche il valore di Na fuori range.

tabella=table(s,h,'RowNames',righe);
```

Cod. 5.2 – Creazione di una tabella riassuntiva dell'esito della ricerca di correlazione.

```
load('Nan_calc.mat')
load('Nan_mis.mat')
load('Range_calc.mat')
load('Range_mis.mat')
tab=zeros(86602,2);

for i=1:86602
   tab(i,1)=sum(range_calc(i,:))+sum(range_mis(i,:)); %la somma dei fuori range per ega
    tab(i,2)=sum(nan_calc(i,:))+sum(nan_mis(i,:)); %la somma dei nan per ega
end

titoli={'Fuori_range_per_EGA' 'Nan_per_EGA'};
tab= array2table(tab,'VariableNames',titoli);
```

Cod. 6 – Script che si serve dei dati grezzi fino ad ora raccolti per creare una table che indica la quantità di valori fuori range e valori NaN per ogni EGA (riga).

```
function [numerr]=NumErr(err)
e=[];
for i=1:length(err)
    e1=string(err(i).errore);
    e=[e;e1];
end % converto gli errori in array di stringhe per la funzione unique
for i=1:length(err)
rig(i,1)=(err(i).riga);
end %creo il vettore riga "paziente"
a=unique(rig); %solo le righe che presentano errori
for i=1:length(a)
    sel=rig==a(i);
    k=find(sel);
    c=length(k);
    u=unique(e(sel))';
    f(i,1)=size(u,2);
    ut(i,1:f(i))=(u);
end
numerr=unique(e);
for i=1:length(numerr)
    d=count(ut,numerr(i));
    s=sum(sum(d));
    numerr(i,2)=s;
end %trova errori per riga

x=double(numerr(:,2));
[x,f]=sort(x,'descend');
y=numerr(:,1);  %ordino
numerr(:,1)=y(f);
numerr(:,2)=x
```

Cod. 7 – La funzione NumErr qui creata prende in ingresso la struttura errori creata in precedenza e ne estrapola una lista ordinata comprensiva di tutti i messaggi di errore dell'apparecchio con relativo numero.

```
load('capillare.mat')
e=fieldnames(capillare);
c=[capillare.SO2];
d=1;
for x=20:5:90  %incrementa da 20 a novanta con incrementi di 5
    sel=c<x;
    sel2=c>x;
    satmeno=capillare(sel,:);
    satpiu=capillare(sel2,:);
    for i=3:36
        a=[satmeno.(char(e(i)))]; %prendo le stesse variabili tra i 2 gruppi
        b=[satpiu.(char(e(i)))];
        [h,p] = ttest2(a,b); %ci interessa il valore p
        val(1,i-2)=p;% che inserisco in un vettore
    end
    ptot(d,:)=val;
    d=d+1;
end
```

Cod. 8 *- t-test recursivo tra campioni capillari separati tramite il valore di saturazione, che aumento da 20 a 90 con incrementi di 5.*

```
load('capillare.mat')
e=fieldnames(capillare);
c=[capillare.SO2];
d=1;
for x=30:5:100
    sel=and(c>=(x-10),c<=(x-5)); %creo i gruppi
    sel2=and(c>(x-5),c<=x);
    satmeno=capillare(sel,:);
    satpiu=capillare(sel2,:);
    for i=3:36
        a=[satmeno.(char(e(i)))]; %prendo le stesse variabili tra i 2 gruppi
        b=[satpiu.(char(e(i)))];
        [h,p] = ttest2(a,b); %ci interessa il valore p
        val(1,i-2)=p;% che inserisco in un vettore
    end
    ptot(d,:)=val;
    d=d+1;
end
```

Cod. 9 - t-test recursivo tra campioni capillari separati in base all'appartenenza ad un intorno della saturazione di ossigeno. Gli intorni comprendono saturazioni comprese in un range del 5% ed a ogni iterazione aumentano del 5%.

```
load('capillare.mat')
e=fieldnames(capillare);
c=[capillare.SO2];
d=1;
for x=40:10:100
    sel=and(c>=(x-20),c<=(x-10)); %creo i gruppi
    sel2=and(c>(x-10),c<=x);
    satmeno=capillare(sel,:);
    satpiu=capillare(sel2,:);
    for i=3:36
        a=[satmeno.(char(e(i)))]; %prendo le stesse variabili tra i 2 gruppi
        b=[satpiu.(char(e(i)))];
        [h,p] = ttest2(a,b); %ci interessa il valore p
        val(1,i-2)=p;% che inserisco in un vettore
    end
    ptot(d,:)=val;
    d=d+1;
end
```

Cod. 10 - t-test recursivo tra campioni capillari separati in base all'appartenenza ad un intorno della saturazione di ossigeno. Gli intorni comprendono saturazioni comprese in un range del 10% ed a ogni iterazione aumentano del 10%.

```
load('Arterioso_scalpo_cord.mat');
load('Capillare.mat');
load('Venoso.mat');

for i = 1:length(venoso.Codice)
    if( strcmp(venoso.PRELIEVO{i},'URINE'))
        a(i)=1;
    else
        a(i)=0;
    end;
end;
Venoso=venoso(not(a),:);
```

Cod. 11 – Esempio di codice per l'eliminazione degli EGA che presentano il valore "URINE" nel campo "PRELIEVO" .

```
load('Arterioso-U.mat');
load('Venoso-U.mat');

Venoso.Tic = 0*ones(length(Venoso.Codice),1);  %0 per venoso
Arterioso.Tic = 1*ones(length(Arterioso.Codice),1); %1 per arterioso(compreso cordone e scalpo
fetale)
t = [Venoso;Arterioso];
t(:,[1 2 5 37:39])= []; %rimuovo le colonne codice, data,anion gap,mod misura,prelievo,tipo
n=height(t);
k = randperm(n);
t=t(k,:); %randomicizzo l'ordine

c = sum(isnan(table2array(t))')';
EGA = t(not(c),:); %rimuove tutte le righe che presentano almeno un NaN

cluster = fitcknn(...
    EGA(:,1:33), ...
    EGA(:,end), ...
    'Distance', 'Euclidean', ...
    'Exponent', [], ...
    'NumNeighbors', 1, ...  %meglio dispari
    'DistanceWeight', 'SquaredInverse', ...
    'Standardize', 1, ...
    'ClassNames',[0; 1]);
```

Cod. 12.1 – Training del clustering per la classificazione degli EGA "Capillari" in "Venosi" e "Arteriosi".

```
load('Capillare-U.mat');
Capillare(:,[1 2 5 37:39])= [];

% e = sum(isnan(table2array(Capillare))')'; %per rimuovere anche i nan
% Capillare = Capillare(not(e),:);

responso=predict(cluster,Capillare);

Capillare.Tic=responso; %con  0=venoso 1=arterioso
b=(Capillare.Tic>0);
capart=Capillare(b,:);
capven=Capillare(not(b),:);

 mArt=mean(capart.SO2, 'omitnan') %media della saturazione di ossigeno per i classificati arteriosi
 mVen=mean(capven.SO2, 'omitnan') %media della saturazione di ossigeno per i classificati venosi

figure
subplot(2,1,1)
histogram(capart.SO2)
title('Saturazione capillari arteriosi')

subplot(2,1,2)
histogram(capven.SO2)
title('Saturazione capillari venosi')

p1 = crossval(cluster);
Eff = 1 - kfoldLoss(p1, 'LossFun', 'ClassifError'); %accuratezza del modello
```

Cod. 12.2 – Test del modello classificatore sugli EGA "Capillari", con istogrammi dei risultati e calcolo dell'accuratezza.

6 Bibliografia e Sitografia

- Radiometer Medical ApS, "Abl700 series reference manual", 2009.
- Angelo Morzenti, Alessandro Campi, Elisabetta Di Nitto, "Introduzione alla programmazione in Matlab", 2011.
- Claudio Rugarli, "Medicina interna sistematica", settima edizione, 2015
- https://www.radiometer.com/
- https://it.mathworks.com/matlabcentral/answers/index