uuuu	a	uvuu	q	buuu	G	bvuu	W	vuuu	6	vvuu	'	puuu	class	pvuu	double
uuub	a	uvub	q	buub	G	bvub	W	vuub	6	vvub	'	puub	class	pvub	double
uuuv	b	uvuv	r	buuv	H	bvuv	X	vuuv	6	vvuv	:	puuv	class	pvuv	double
uuup	b	uvup	r	buup	H	bvup	X	vuup	6	vvup	:	puup	class	pvup	double
uubu	c	uvbu	s	bubu	I	bvbu	Y	vubu	7	vvbu	++	pubu	interface	pvbu	true
uubb	c	uvbb	s	bubb	I	bvbb	Y	vubb	7	vvbb	++	pubb	interface	pvbb	true
uubv	d	uvbv	t	bubv	J	bvbv	Z	vubv	7	vvbv	--	pubv	interface	pvbv	true
uubp	d	uvbp	t	bubp	J	bvbp	Z	vubp	7	vvbp	--	pubp	interface	pvbp	true
uuvu	e	uvvu	u	buvu	K	bvvu	0	vuvu	8	vvvu	;	puvu	static	pvvu	false
uuvb	e	uvvb	u	buvb	K	bvvb	0	vuvb	8	vvvb	;	puvb	static	pvvb	false
uuvv	f	uvvv	v	buvv	L	bvvv	0	vuvv	8	vvvv	\n	puvv	static	pvvv	false
uuvp	f	uvvp	v	buvp	L	bvvp	0	vuvp	8	vvvp	\n	puvp	static	pvvp	false
uupu	g	uvpu	w	bupu	M	bvpu	1	vupu	9	vvpu	\t	pupu	final	pvpu	if
uupb	g	uvpb	w	bupb	M	bvpb	1	vupb	9	vvpb	\t	pupb	final	pvpb	if
uupv	h	uvpv	x	bupv	N	bvpv	1	vupv	9	vvpv	\s (space)	pupv	final	pvpv	else
uupp	h	uvpp	x	bupp	N	bvpp	1	vupp	9	vvpp	\s (space)	pupp	final	pvpp	else
ubuu	i	upuu	y	bbuu	O	bpuu	2	vbuu	=	vpuu	(	pbuu	private	ppuu	while
ubub	i	upub	y	bbub	O	bpub	2	vbub	=	vpub	(	pbub	private	ppub	while
ubuv	j	upuv	z	bbuv	P	bpuv	2	vbuv	!	vpuv	)	pbuv	private	ppuv	while
ubup	j	upup	z	bbup	P	bpup	2	vbup	!	vpup	)	pbup	private	ppup	while
ubbu	k	upbu	A	bbbu	Q	bpbu	3	vbbu	.	vpbu	[	pbbu	public	ppbu	do
ubbb	k	upbb	A	bbbb	Q	bpbb	3	vbbb	.	vpbb	[	pbbb	public	ppbb	do
ubbv	l	upbv	B	bbbv	R	bpbv	3	vbbv	-	vpbv	]	pbbv	public	ppbv	do
ubbp	l	upbp	B	bbbp	R	bpbp	3	vbbp	-	vpbp	]	pbbp	public	ppbp	do
ubvu	m	upvu	C	bbvu	S	bpvu	4	vbvu	+	vpvu	{	pbvu	void	ppvu	return
ubvb	m	upvb	C	bbvb	S	bpvb	4	vbvb	+	vpvb	{	pbvb	void	ppvb	return
ubvv	n	upvv	D	bbvv	T	bpvv	4	vbvv	/	vpvv	}	pbvv	void	ppvv	return
ubvp	n	upvp	D	bbvp	T	bpvp	4	vbvp	/	vpvp	}	pbvp	void	ppvp	return
ubpu	o	uppu	E	bbpu	U	bppu	5	vbpu	*	vppu	<	pbpu	int	pppu	[STOP]
ubpb	o	uppb	E	bbpb	U	bppb	5	vbpb	*	vppb	<	pbpb	int	pppb	[STOP]
ubpv	p	uppv	F	bbpv	V	bppv	5	vbpv	"	vppv	>	pbpv	int	pppv	[STOP]
ubpp	p	uppp	F	bbpp	V	bppp	5	vbpp	"	vppp	>	pbpp	int	pppp	[STOP]

<= Quadruplets

Triplets =>

aaa	a	taa	2
aac	b	tac	2
aat	c	tat	2
aag	d	tag	3
aca	e	tca	3
acc	f	tcc	3
act	g	tct	4
acg	h	tcg	4
ata	i	tta	4
atc	j	ttc	5
att	k	ttt	5
atg	l	ttg	5
aga	m	tga	6
agc	n	tgc	6
agt	o	tgt	6
agg	p	tgg	7
caa	q	gaa	7
cac	r	gac	7
cat	s	gat	8
cag	t	gag	8
cca	u	gca	8
ccc	v	gcc	9
cct	w	gct	9
ccg	x	gcg	9
cta	y	gta	:
ctc	z	gtc	:
ctt	0	gtt	'
ctg	0	gtg	'
cga	0	gga	M
cgc	1	ggc	M
cgt	1	ggt	*
cgg	1	ggg	*

Triplets pour les codes gènes simples (cf. Creatures) ; Qudruplets pour les codes "executables".

Fiche indicatrice des codes de triplets / quadruplets pour encodage des génomes *Silico*